

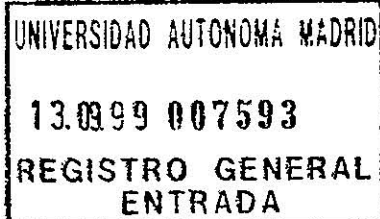
T/334

GEDEON: GErador DE Oraciones en lenguaje Natural

Una aproximación computacional a los procesos de codificación gramatical

AUTORA: Pei-Chun Shih Ma

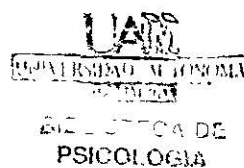
DIRECTORES: Dra. Mercedes Belinchón Carmona
Dr. Pablo Adarraga Morales



Madrid, Septiembre de 1999

R. p. - 21.914

TESIS DOCTORAL



DEPARTAMENTO DE PSICOLOGÍA BÁSICA
FACULTAD DE PSICOLOGÍA
UNIVERSIDAD AUTÓNOMA DE MADRID
Cantoblanco, Madrid

UNIVERSIDAD
AUTONOMA DE
MADRID
BIBLIOTECA

R.B.C. 68.538.

Agradecimientos

Cierto es, que escrito está, que "Más bienaventurado es dar que recibir". Sin embargo, no puedo dejar de reconocer el haber disfrutado de ser la receptora, durante esta travesía particular, de las acciones generosas de otros, algunas de ellas pequeñas muestras de aprecio y muchas de ellas grandes demostraciones de amistad. Para ellos va aquí una pequeña muestra de mi profunda gratitud.

A los directores de esta Tesis. A Mercedes Belinchón, mi Obi-Wan Kenobi particular, por todos estos años en los que me ha guiado en el quehacer científico y, en ocasiones, no sólo en eso. A Pablo Adarraga, a quien debo muchos de los conocimientos adquiridos sobre el buen hacer científico.

A los directores virtuales de esta Tesis. A Eduard Hovy, cuyo trabajo ha sido de inspiración para el mío y cuya persona ha sido de estímulo para mí. A José Santacreu por su buen sentido de la orientación, marcando el rumbo que debo llevar.

A María Oliva Márquez, por las 100 razones y, una directamente relacionada con el hecho de que esta Tesis haya visto la luz. A Alberto Becerra que ha contribuido con su especial forma de ser a mi formación en general, lecciones recibidas siempre entre algún refrigerio. A José Luis Zaccagnini por haberme introducido en el apasionante mundo de la inteligencia artificial.

A Víctor Rubio (padre) por su excelente trabajo de análisis sintáctico con el que me facilitó la vía para escapar del callejón sin salida en el que me encontraba. A Juan Antonio Talavera por haber escuchado con interés los sucesivos progresos producidos en la realización de este trabajo. A Amparo Caballero y Flor Sánchez por su constante preocupación sobre mi bienestar psicológico.

A Lázaro Cobiella, Dolores Muñoz, Yolanda Postigo y Sara Ulla (en orden alfabético, para que nadie se ofenda) por todas las huellas que han dejado a lo largo de la Tesis y una razón adicional. A Lázaro por ese filamento de sensatez que hay en el fondo de ese mar de locura caribeña. A Lola por nuestra amistad. A Yolanda por su paciencia oriental con quien carece de ella. A Sara por todos los buenos ratos que hemos pasado juntas.

A Oscar García por ser mi copiloto favorito, no sólo en la carretera. A María José Contreras por las pinceladas de color (en especial, Azul) que ha introducido a mi mundo personal. A Pablo de Felipe por esas largas charlas sobre los mares y los peces.

A mis amigos fuera del mundo académico, Dora, Gerty, Javi, Juan, Miriam, Yolanda y Violeta (en orden alfabético), por ser los sufridores tanto de mis largas ausencias como de mis breves presencias. A Juan Andrés, mi sobrino postizo favorito, por la alegría que me produce el estar con él.

Y a Tí, Padre amoroso, Amigo fiel y Compañero en el andar diario, gracias.

Índice de contenido

Introducción	1
Capítulo 1: Marco teórico y metodológico	7
1.1. Introducción	7
1.2. Psicología cognitiva del procesamiento de la información	9
1.2.1. Principios básicos	11
1.2.1.1. Mentalismo	11
1.2.1.2. Funcionalismo	11
1.2.1.3. Presupuesto computacional	12
1.2.2. Paradigma computacional o de procesamiento de la información	12
1.3. Implicaciones de la inteligencia artificial en la construcción teórica de la psicología	13
1.3.1. Lenguajes de programación	15
1.3.2. Simulación	25
Capítulo 2: Procesos de producción del lenguaje	31
2.1. Introducción	31
2.2. Principales fuentes de datos empíricos	33
2.3. Caracterización funcional del sistema de producción del lenguaje	39
2.4. Codificación gramatical	42
2.5. Recapitulación.....	48
Capítulo 3: Procesamiento de lenguaje natural	51
3.1. Introducción	51
3.2. Procesamiento de lenguaje natural	52
3.3. Técnicas de codificación y utilización de información gramatical	54
3.3.1. Gramáticas libres de contexto	55
3.3.2. Gramática transformacional	58
3.3.3. Redes de transición aumentada	58
3.3.4. Unificación	61
3.3.5. Gramáticas léxico-sintagmáticas	63
3.3.6. Gramática sistémica funcional	65
3.4. Sistemas de generación táctica	70
3.4.1. MARGIE-BABEL	71
3.4.2. COMET	71
3.4.3. PAULINE	72
3.4.4. PENMAN	75
3.5. Recapitulación	79

Capítulo 4: Sistema de generación de oraciones GEDEON	81
4.1. Introducción	81
4.2. Descripción del sistema	83
4.2.1. Arquitectura del programa	84
4.2.1.1. <i>Upper-Model</i>	85
4.2.1.1.1. La subjerarquía <i>PROCESS</i>	86
4.2.1.1.2. La subjerarquía <i>OBJECT</i>	91
4.2.1.1.3. La subjerarquía <i>QUALITY</i>	92
4.2.1.2. <i>Domain-Model</i>	93
4.2.1.3. <i>Lexicon</i>	95
4.2.1.3.1. La jerarquía <i>Lexicon</i>	95
4.2.1.3.2. La jerarquía <i>Affix</i>	97
4.2.1.4. <i>Grammar</i>	99
4.2.1.5. <i>Goals</i>	101
4.2.1.6. <i>Input</i>	104
4.2.1.7. <i>Output</i>	106
4.2.1.8. <i>Main-Process</i>	107
4.2.2. Funcionamiento del sistema	108
4.2.3. Un ejemplo comentado de realización	109
4.3. Procedimiento de desarrollo	115
 Capítulo 5: Evaluación y validación del sistema	119
5.1. Introducción	119
5.2. Estudio 1	120
5.2.1. Método	121
5.2.1.1. Sujetos	121
5.2.1.2. Procedimiento	121
5.2.1.3. Criterios de análisis	124
5.2.2. Resultados	124
5.2.2.1. Análisis descriptivo	124
5.2.2.3. Contraste de hipótesis	125
5.2.3. Conclusiones	126
5.3. Estudio 2	126
5.3.1. Método	128
5.3.1.1. Procedimiento	128
5.3.1.2. Criterios de análisis	128
5.3.2. Resultados	129
5.3.2.1. Análisis descriptivo	129
5.3.2.2. Contraste de hipótesis	130
5.3.3. Conclusiones	132
 Capítulo 6: Conclusiones generales	135
 Referencias bibliográficas	141
 Anexo: Manual de Usuario	149

Introducción

En el desarrollo de la ciencia moderna intervinieron dos factores fundamentales que caracterizaron la labor de los filósofos científicos del Renacimiento. Por un lado, la importancia que, a partir de los nominalistas, habían adquirido la observación y la experimentación; y por otro, el descubrimiento de la tradición pitagórica, que contribuyó a la consolidación de la concepción de los fenómenos físicos en términos matemáticos. Esta insistencia en presentar los descubrimientos empíricos en el lenguaje de las matemáticas y el empeño en conciliar la física y el cálculo era la constante compartida por los primeros científicos modernos como Copérnico, Kepler y Galileo en cuyas obras se observa la seña de identidad de una nueva era: fundir la observación empírica y la especulación teórica (Navarro y Calvo, 1983).

Más recientemente, en el estudio de los fenómenos cognitivos, el enfoque del procesamiento de la información ha sido considerado como una alternativa, de entre los distintos paradigmas de la psicología cognitiva, que ofrece la precisión y la confirmabilidad exigida por la psicología científica (Boden, 1977), conjugando la investigación empírica de los fenómenos cognitivos basada, principalmente, en la metodología experimental, con el desarrollo teórico fundamentado en la metáfora del ordenador. Sin embargo, la controversia suscitada en torno a la equivalencia funcional establecida entre la cognición y el ordenador dio lugar a la distinción entre una versión débil y una versión fuerte de esta metáfora (véase de Vega, 1982), limitando así, en gran medida, las posibilidades de desarrollo teórico de la psicología cognitiva del procesamiento de la información.

La psicología cognitiva del procesamiento de la información se ha caracterizado por incorporar elementos conceptuales y formales de las ciencias de la computación como herramientas de apoyo para la descripción de los datos empíricos obtenidos. Sin embargo, desde la versión débil, negando la equivalencia funcional, fundamento de la aplicación de la ciencia de la computación a la psicología, el uso de estos elementos ha sido generalmente impreciso y, en algunos casos, erróneo. Un ejemplo de este hecho es el uso *adaptado* de los "diagramas de flujo" como medio para representar modelos psicológicos de los procesos cognitivos, práctica muy común entre los psicólogos del enfoque del procesamiento de la información (de Vega, 1982, 1984). El problema que plantea esta estrategia es que al no respetar las prescripciones

establecidas para su uso, introducen ambigüedades sintácticas y semánticas en sus representaciones.

Por su parte, la inteligencia artificial, en su afán de describir computacionalmente los comportamientos "inteligentes" observados en las personas, ha desarrollado, de forma racional, formalismos orientados a la expresión de los procesos cognitivos así como sistemas que reproducen algunas de estas conductas (Newell y Simon, 1972; Anderson, 1983; Newell, 1990). En esto, la inteligencia artificial constituye un tipo de psicología teórica que aborda un objeto de estudio común a la psicología: los fenómenos cognitivos (Collins y Smith, 1988). Así, tal y como exponen estos autores, la interacción entre la psicología, fundamentalmente empírica, y la inteligencia artificial, fundamentalmente racional, constituye una estrategia sólida para el desarrollo teórico propio de la psicología cognitiva.

Por tanto, sin negar las acertadas observaciones realizadas sobre las limitaciones de la metáfora del ordenador (de Vega, 1982), consideramos que:

1. La precisión requerida en las teorías científicas hace que los lenguajes naturales, esencialmente ambiguos, sean poco adecuados para una rigurosa formalización científica de las observaciones empíricas. Si bien los lenguajes de programación no gozan del estatus de las matemáticas para la descripción de los fenómenos físicos, sí se ofrecen como una alternativa apropiada para la descripción de los fenómenos cognitivos.
2. El uso de la metodología de la simulación no debe restringirse únicamente a la modelización de teorías psicológicas bien definidas. Esta exigencia, no observada en otras estrategias de investigación utilizadas en psicología, limita la capacidad de la simulación en su contribución al desarrollo teórico.

El objetivo de este trabajo es recorrer el puente que concilia la psicología y la inteligencia artificial, aplicando las herramientas de formalización, desarrolladas en la inteligencia artificial, en la construcción teórica de la psicología. Asimismo, este trabajo se ofrece como un espacio en el que la metodología de la simulación tiene una oportunidad para expresar su utilidad para la investigación en psicología. Con todo, avanzar un poco más en el camino de la integración de la rama teórica y la rama empírica del estudio de los fenómenos cognitivos.

Es, por tanto, un aspecto esencial para este trabajo, la construcción de un sistema de inteligencia artificial que reproduce un fenómeno cognitivo específico en cuyo proceso de desarrollo participen las consideraciones teóricas observadas en ambas disciplinas, la psicología y la inteligencia artificial.

El fenómeno cognitivo considerado, nuestro objeto de estudio es la codificación gramatical. Sobre esta cuestión, la investigación psicolingüística ha generado un amplio conjunto de datos empíricos que han contribuido a la elaboración de esquemas teóricos acerca de los procesos y las representaciones que participan en la construcción de las oraciones (Garrett, 1988; Levelt, 1989; Bock y Levelt, 1994; Igoa y García-Albea, 1999). Asimismo, en la inteligencia artificial, lejos ya de los primeros sonados fracasos de la traducción mecánica (Booth, 1948; Reifler, 1954; Weaver, 1955; Oettinger, 1957; Bar-Hillel, 1960), observamos grandes avances tecnológicos en el área del procesamiento de lenguaje natural en el que se han desarrollado herramientas conceptuales y metodológicas sólidas y bien definidas (Thompson, 1977; Reiter, 1994).

Así, conjugando las aportaciones realizadas desde ambas disciplinas, hemos procedido a desarrollar un sistema de generación de oraciones individuales que constituye, por un lado, el componente fundamental de nuestro trabajo de simulación y, por otro lado, la base para una formalización de los procesos de codificación gramatical.

Sin duda, las implicaciones que un sistema de simulación representa para la construcción teórica de la psicología es una cuestión que requiere de un análisis empírico. Esto es, para la psicología, no es suficiente contar con un sistema de inteligencia artificial que produzca oraciones en lenguaje natural, sino que en su ejecución, puedan observarse las operaciones que intervienen en la producción de oraciones de sujetos reales. Es imprescindible, por tanto, en cualquier investigación de simulación, contar con análisis empíricos del comportamiento del sistema de simulación de manera que sean comparables los datos obtenidos con los de los sujetos reales.

Con este propósito, realizamos un análisis empírico del comportamiento de nuestro sistema en el que observamos el grado en que alcanza los criterios exigidos a los sistemas de simulación (Fodor, 1968; García-Albea, 1981). Los datos obtenidos en este análisis han determinado las consecuencias teóricas que de nuestro trabajo de simulación se derivan.

Probablemente, la realización de este trabajo no supone importantes aportaciones al desarrollo teórico en la psicolingüística, si bien contribuye al esclarecimiento de algunas de las cuestiones en ella planteadas. En esto, consideramos que este trabajo constituye una demostración de que la construcción teórica de la psicología del procesamiento de la información debe fundamentarse en un constante ir y venir entre la psicología y la inteligencia artificial. Ello constituye un estilo del quehacer científico caracterizado por la formalización computacional de los fenómenos cognitivos en la misma línea que los fenómenos físicos se expresan en términos matemáticos.

Nuestro planteamiento no hace más que retomar la perspectiva inicial del enfoque de procesamiento de la información, relegada durante muchos años a un segundo plano por las controversias suscitadas en torno al presupuesto computacional. Hoy día, la reconciliación entre la psicología y la ciencia de la computación, cada vez más frecuente entre los psicólogos cognitivos (véase Bock y Levelt, 1994; Igoa y García-Albea, 1999), sigue provocando polémica (*e.g.*, *Seminario: Perspectiva actuales en Psicología Cognitiva. Madrid, 1998*). Sin embargo, consideramos que este trabajo, entre otros (*i.e.*, Roelofs, 1992; Dell y Burger, 1997), es un ejemplo de que *el movimiento se demuestra andando*.

Vamos a iniciar la exposición del trabajo realizado con la presentación del marco teórico y metodológico en el que se inscribe esta Tesis. Comenzaremos con una breve descripción del paradigma de la psicología cognitiva del procesamiento de la información, haciendo especial énfasis en las implicaciones de la inteligencia artificial en la construcción teórica de la psicología.

Los dos capítulos siguientes están dedicados a la exposición de cuestiones que están directamente relacionado, por un lado, con nuestro objeto de estudio, la codificación gramatical, y, por otro lado, con el objetivo de este trabajo, la construcción de un sistema de simulación. Así, en el segundo capítulo, presentaremos las principales propuestas teóricas realizadas desde la psicolingüística, describiendo con ello el estado actual en el que se encuentra la elaboración de una teoría psicológica acerca de los procesos implicados en la codificación gramatical.

En el tercer capítulo, vamos a exponer las aportaciones realizadas desde la inteligencia artificial. Describiremos las principales estrategias desarrolladas por investigadores del procesamiento de lenguaje natural para la construcción de generadores automáticos de oraciones

individuales. Asimismo, presentaremos algunos de los sistemas de generación de oraciones en lenguaje natural más representativos en la historia de este área de la inteligencia artificial.

Los capítulos cuarto y quinto están dedicados a la exposición del trabajo empírico realizado para esta Tesis. En el capítulo cuarto, presentaremos nuestro sistema de generación de oraciones en lenguaje natural, el sistema GEDEON, desarrollado como el elemento básico de una investigación en la que se aplica la metodología de la simulación para el análisis de su objeto de estudio. Describiremos, por un lado, los componentes y el funcionamiento de nuestro sistema y, por otro lado, las distintas tareas realizadas para su construcción. En el capítulo quinto, presentaremos los estudios realizados para la evaluación de nuestro sistema en base a los criterios establecidos para la valoración de sistemas de simulación.

Finalmente, en el último capítulo, expondremos las conclusiones derivadas, a la vista de los resultados obtenidos, acerca de nuestro objeto de estudio en particular, los procesos de codificación gramatical, y las implicaciones de este trabajo para la psicología en general.

Capítulo 1: Marco teórico y metodológico

1.1. Introducción

La capacidad humana para comunicarse por medio de un código simbólico arbitrario como el lenguaje verbal es un fenómeno tan cotidiano como complejo. Es una habilidad que ejercitamos a diario en las más diversas situaciones y condiciones, aparentemente sin grandes esfuerzos. Sin embargo, no nos es extraño encontrarnos alguna vez en la situación de decidir qué decir, de cómo decir algo o de contabilizar a algún profesor más de cien “ees” durante una hora de clase. Observemos durante un momento el siguiente diálogo que nos resultará bastante familiar:

A: ¿Tiene hora?

B: No llevo reloj.

En esta situación tenemos a un sujeto A que ante la necesidad de conocer una información específica decide formular una pregunta a su interlocutor, el sujeto B, quien, a su vez, decide informarle de su incapacidad para proporcionar la información solicitada. Todo esto, en apariencia no parece requerir ningún esfuerzo.

Ahora bien, ¿cuáles son las operaciones que el sujeto A realiza para formular una frase que exprese su deseo de saber la hora, o sobre qué elementos opera el sujeto B para extraer de dicha frase el deseo de su interlocutor? Éstas son algunas de las cuestiones que los investigadores de la psicología del lenguaje intentan explicar.

La psicología del lenguaje como parte de la psicología científica define su objeto de estudio, desarrolla sus investigaciones empíricas y formula sus modelos teóricos dentro de los parámetros establecidos por los distintos paradigmas teóricos de la psicología científica. Así, las perspectivas adoptadas por los investigadores acerca de la función psicológica del lenguaje han ido evolucionando a la par que la psicología científica desde su inicio.

Esta variabilidad de enfoques responde al hecho de que no existe un único modo, científicamente riguroso, de describir y explicar la actividad lingüística. Al igual que otros tipos de comportamiento, en especial aquellos etiquetados como inteligentes, la actividad lingüística puede ser estudiada desde diferentes perspectivas. Estas responden a los cuatro niveles básicos de descripción: el físico o neurobiológico, el conductual, el intencional y el computacional (Belinchón, Rivièr e Igoa, 1992).

Al nivel físico o neurobiológico le corresponde el análisis sistemático del sustrato material en que se asientan tanto la capacidad humana para el lenguaje como la actividad lingüística efectiva. La identificación de las estructuras cerebrales que participan en comportamientos lingüísticos concretos mediante el análisis sistemático de la actividad cerebral y las deficiencias asociadas a lesiones de zonas cerebrales específicas, son ejemplos del quehacer de los investigadores que adoptan esta perspectiva de caracterización científica de la actividad lingüística.

En el nivel conductual, lo que interesa son las acciones directamente observables. Para su descripción del comportamiento lingüístico manifiesto del sujeto, hace uso de un lenguaje típicamente observacional que descansa sobre un entramado teórico y conceptual que no puede ser reducido a los aspectos neurofisiológicos ni debe hacer referencia alguna a constructos mentalistas. La topografía, frecuencia o intensidad de las respuestas lingüísticas, así como sus antecedentes y consecuentes en su entorno de emisión, son algunos de los términos en los que se describe la actividad lingüística a este nivel.

Desde la perspectiva intencional, el lenguaje se define como un instrumento de representación de la realidad y de comunicación interpersonal. Se caracteriza, a diferencia de los anteriores, por otorgarle al sujeto un papel activo y por la utilización de un vocabulario mentalista que hace referencia a atribuciones psicológicas o predicados mentales relativos a las ideas, las creencias, los deseos o las expectativas de los sujetos que producen o comprenden los mensajes lingüísticos.

Por último, el nivel computacional se caracteriza por definir la actividad lingüística como un conjunto de procesos mentales de cómputo que operan sobre representaciones de tipo simbólico. Esta perspectiva procesual del lenguaje estaba ya presente en los trabajos de Wundt (Blumenthal, 1970). Sin embargo, las limitaciones del método de introspección, especialmente evidentes en estos procesos que operan, con frecuencia, por debajo del nivel de la conciencia de

los sujetos, ha hecho que esta línea de investigación en la psicología del lenguaje estuviese aparcada hasta la consolidación de la psicología cognitiva del procesamiento de la información. Este enfoque de la psicología científica ha proporcionado a los investigadores de la actividad lingüística las herramientas teóricas y metodológicas para describir los procesos psicolingüísticos implicados en la producción y comprensión del lenguaje.

Esta última perspectiva, la computacional, es la adoptada en este trabajo. Teniendo en cuenta que las herramientas teóricas y metodológicas son proporcionadas por la psicología cognitiva del procesamiento de la información, consideramos conveniente, e incluso necesario, detenernos con cierto detalle en la revisión de este paradigma de la psicología científica antes de pasar a exponer aspectos más específicos de nuestro objeto de estudio: los procesos de producción del lenguaje.

1.2. Psicología cognitiva del procesamiento de la información

La psicología, en su afán de alcanzar el reconocimiento como disciplina científica, limitó el análisis de su objeto de estudio a fenómenos observables y objetivables. Este reduccionismo, característico del conductismo, es fundamentalmente metodológico. Esto es, el rechazo de aspectos mentales para la explicación de la conducta se debe al presupuesto de que son cuestiones que no pueden ser tratadas de manera científicamente aceptable. Sin embargo, tras varias décadas de dominancia del paradigma conductista en la psicología científica, la necesidad de ampliar su marco de referencia a elementos explicativos de la conducta, más allá de las características físicas observables, se fue haciendo cada vez más evidente (Zaccagnini y Delclaux, 1982). La conducta lingüística es un buen ejemplo de ello.

Sin dejar de reconocer el claro valor aplicado del análisis skinneriano de la conducta verbal, sus insuficiencias fueron explicitadas en las críticas formuladas por Chomsky a su explicación del lenguaje fundamentada en la descripción de las secuencias de estimulación y comportamiento (Chomsky, 1959a). En primer lugar, hay fenómenos lingüísticos observables, como los mensajes funcionalmente equivalentes pero formalmente alternativos o los errores en la conjugación de verbos irregulares por la hiperregularización, que denotan aspectos esenciales del lenguaje como son su carácter creativo y sistemático, que no son explicables desde esta interpretación reduccionista de la actividad lingüística. Es más, Chomsky trasladó a la psicología su insatisfacción por las teorías lingüísticas del momento. Así, una teoría explicativa

del lenguaje no sólo tiene que dar cuenta de la conducta lingüística efectiva de los sujetos, sino también, de su conducta potencial. Para ello, es imprescindible concebir el lenguaje como una capacidad que puede ser descrita con independencia de las conductas lingüísticas concretas que ejecutan los individuos particulares. Esta distinción entre la actuación y la competencia lingüística exige, de la lingüística, una descripción abstracta del conocimiento lingüístico en términos de reglas gramaticales, más allá de los inventarios y clasificaciones de elementos lingüísticos, y de la psicología, una explicación de la capacidad lingüística humana en términos de manejo de reglas, más allá de las relaciones de contingencia entre estímulos y respuestas.

La necesidad de recurrir a procesos internos para explicar conductas complejas, así como la posibilidad de su tratamiento científico, estaban también presentes en trabajos de otras disciplinas. Wiener (1948) introdujo el concepto de retroalimentación para explicar el funcionamiento de los servomecanismos que mantienen el rumbo de los proyectiles de artillería antiaérea, misiles guiados y aviones. Estudió y modeló matemáticamente estos mecanismos de control, y apoyándose en la teoría de la información desarrollada por Shannon (1948), propuso un nivel particular de descripción para referirse a los sistemas que cuentan con un proceso circular de evaluación y corrección de su comportamiento en función de su meta. Esto es, la información como entidad conceptual independiente de la materia y la energía, es decir, independiente de todo dispositivo concreto de transmisión y de su contenido. Ésta es la noción clave de la teoría de la información. En ella, Shannon identificó el concepto de información con la idea de reducción de incertidumbre y propuso una medida matemática, el bit, como su unidad básica. Cada bit contiene la cantidad de información requerida para seleccionar una opción entre dos alternativas equiprobables. Así, para seleccionar una opción de entre ocho alternativas equiprobables se requiere tres unidades de información, tres bits. El primer bit limita las alternativas a cuatro, de las que el segundo bit descarta dos, para que con el tercer bit se seleccione una de las dos alternativas restantes. Con ello, no sólo demostró la posibilidad de formalización científica de conceptos “mentalistas” como la información, sino que, equiparándolo con otros sistemas duales como la lógica proposicional y los circuitos de conmutación, sentó las bases para el desarrollo de los ordenadores digitales.

Finalmente, el desarrollo de los ordenadores digitales y el nacimiento de la inteligencia artificial han proporcionado a los investigadores herramientas conceptuales y metodológicas para abordar de forma científica los fenómenos mentales. El ordenador, tanto por su capacidad de almacenamiento y manejo de información como por su capacidad de generación de conductas inteligentes, se ofrece como un posible modelo de la mente humana. Los

investigadores de la inteligencia artificial, como Newell y Simon, en su pretensión de construir programas de ordenador en los que se observen capacidades semejantes a las funciones cognitivas, han proporcionado a la psicología científica una nueva metodología para el estudio de los fenómenos mentales: la simulación.

Todo ello ha contribuido a la consolidación del enfoque de procesamiento de la información como paradigma de la psicología cognitiva.

1.2.1. Principios básicos

La psicología cognitiva del procesamiento de la información como disciplina científica cuenta con un conjunto de metapostulados fundamentales que definen su objeto de estudio y especifican el modelo de ciencia a seguir. Éstas son: el mentalismo, el funcionalismo y el presupuesto computacional (de Vega, 1984; Adarraga, 1991).

1.2.1.1. Mentalismo

Este presupuesto no sólo reconoce la existencia de los fenómenos mentales, sino que les concede un papel causal en la generación de la conducta inteligente. A su vez, las propiedades que definen tales fenómenos son propiedades emergentes intrínsecamente no reductibles al funcionamiento del sustrato físico. Esto hace imprescindible el establecimiento de un nivel particular de descripción: el nivel de la representación (Pylyshyn, 1984). Esto es, el funcionamiento cognitivo debe ser descrito en función de símbolos, esquemas, imágenes, ideas y otras formas de representación mental, entendidos como sistemas formales que hacen explícitas ciertas entidades o tipos de información junto con especificaciones de su elaboración y utilización. Así, la explicación de los fenómenos mentales debe realizarse en función del papel que cada uno de ellos desempeña en la causación de la actividad intelectual, partiendo de una descripción de dichos fenómenos en términos representacionales.

1.2.1.2. Funcionalismo

Tradicionalmente, el mentalismo se ha asociado y, a veces, identificado con el dualismo. La propuesta del funcionalismo reconoce la independencia de la cognición respecto a las características específicas de los sistemas que le sirven de sustrato, sin, por ello, considerar la

mente como una sustancia distinta de lo físico, sino definida por las propiedades funcionales que emergen del tipo peculiar de organización de la materia (Fodor, 1981).

1.2.1.3. Presupuesto computacional

Desde el punto de vista representacional, la mente se define como un sistema de manipulación de símbolos o representaciones (Newell, 1980). Esta capacidad, resultado de la organización funcional de la mente, es definida como un conjunto de funciones de procesamiento de la información (Newell, Shaw y Simon, 1964; Pylyshyn, 1984). Lo que el presupuesto computacional establece es una descripción de las funciones cognitivas en términos de procesos de cómputo y, como tales, se sirven de los ordenadores para entender la mente (Pylyshyn, 1989). Esto es, la distribución y la organización del ordenador como sistema de manipulación de símbolos, es un modelo útil para el estudio del sistema cognitivo y sus procesos. Es más, la simulación de fenómenos cognitivos es considerada como una fuente de datos para la verificación empírica en la psicología.

1.2.2. Paradigma computacional o de procesamiento de la información

Considerando lo anterior, podemos decir que la psicología cognitiva del procesamiento de la información analiza los fenómenos cognitivos en términos de representaciones y procesos mentales. Estos deben ser definidos en función de los papeles que desempeñan en la causación de la conducta inteligente y no según su contenido ni su naturaleza o constitución física. Es en este sentido, la concepción de la mente humana, a la vez, como un sistema físico y como un mecanismo abstracto de computación o de procesamiento de la información, donde el ordenador resulta una herramienta provechosa para el estudio de la cognición. A un nivel abstracto, es decir, con independencia de sus propiedades físicas, el sistema cognitivo humano y los programas de inteligencia artificial son casos particulares de una misma clase, la clase de los sistemas de cómputo. Y como tales, se caracterizan por ser *"sistemas cuyas funciones de intercambio con el medio se describen óptimamente como funciones de manejo sistemático de información, [es decir, funciones] por las cuales la información es tomada, almacenada, transformada y retornada al entorno"* (Adarraga, 1994).

Es más, como disciplina científica, el objetivo de la psicología cognitiva es la creación de modelos teóricos de su objeto de estudio: las funciones cognitivas. Estos modelos teóricos deben ser útiles para interpretar de forma unívoca los datos obtenidos mediante investigaciones

empíricas. Para ello, deben estar expresados explícitamente por medio de lenguajes cuyas reglas sean enteramente precisas, de manera que les confieran la capacidad de generar un rango definido, y no otros, de predicciones, fenómenos y comportamientos. Así, los lenguajes de programación utilizados por los sistemas de inteligencia artificial se ofrecen como formalismos aptos para cumplir en la psicología un papel próximo al que el cálculo cumple tradicionalmente en la física (Adarraga, 1993).

Las funciones cognitivas así explicitadas son implementables en los ordenadores, lo que les permite a los investigadores experimentar con los modelos teóricos desarrollados para explicar algún fenómeno psicológico real. La metodología de simulación junto con la experimentación, conforman las principales herramientas de investigación de la psicología cognitiva (Zaccagnini y Delclaux, 1982; Zaccagnini, 1994).

Sin embargo, el objeto de estudio de la psicología cognitiva no es exclusivo de la psicología científica. Es más, dadas las condiciones teóricas que debe cumplir, tampoco puede ser abordado con independencia de otras ciencias. Por su importancia para este trabajo y para la psicología, pasaremos a continuación, a analizar algunos aspectos que hacen de la inteligencia artificial una disciplina complementaria y, en algunos aspectos, imprescindible para la construcción teórica de la psicología.

1.3. Implicaciones de la inteligencia artificial en la construcción teórica de la psicología

Ciertamente, la idea de crear máquinas pensantes no es, en absoluto, novedosa. Sin embargo, la aparición de los ordenadores electrónicos supuso el avance tecnológico necesario para su instrumentalización. La inteligencia artificial nace como disciplina científica que explora las posibilidades de producir programas de ordenador capaces de comportarse inteligentemente.

Este supuesto básico en el que se fundamenta las investigaciones sobre inteligencia artificial es formalizado en lo que Newell y Simon (1976) llaman la hipótesis del sistema de símbolos físicos. La definición de un sistema de símbolos físicos es:

"Un sistema de símbolos físicos consiste en un conjunto de entidades, llamadas símbolos, que son patrones físicos que pueden funcionar como componentes de otro tipo de entidad llamada expresión (o estructura de símbolos). De esta forma, una estructura de símbolos está compuesta por un número de instancias (señales o tokens) de símbolos relacionados de alguna forma física (como que una señal debe seguir a otra). En algún instante, el sistema contendrá una colección de estas estructuras de símbolos. Además de estas estructuras, el sistema contiene también una colección de procesos que operan sobre expresiones para producir otras expresiones: procesos de creación, modificación, reproducción y destrucción. Un sistema de símbolos físicos es una máquina que produce a lo largo del tiempo una colección evolutiva de estructuras de símbolos. Este sistema existe en un mundo de objetos tan extenso como sus propias expresiones simbólicas."

y el enunciado de la hipótesis es:

"Un sistema de símbolos físicos posee los medios necesarios y suficientes para realizar una acción inteligente genérica."

Fueron estos mismos autores, los primeros investigadores de la inteligencia artificial que, más allá de reflexiones teóricas, demostraron empíricamente la capacidad de los programas de ordenador para generar comportamientos que, en caso de ser expuestos por seres humanos, se considerarían inteligentes. Es más, el Teórico Lógico es un buen ejemplo del papel de la inteligencia artificial en el estudio psicológico de la cognición.

El Teórico Lógico es un programa de demostración de teoremas lógicos (véase Gardner, 1985). Dada una expresión lógica, el objetivo del programa es descubrir cuál es su demostración. Para ello cuenta con un conjunto de reglas básicas de operación y una lista de axiomas y de teoremas ya demostrados. La eficiencia del programa era notable, logrando demostrar los 38 primeros teoremas de los 52 expuestos en el capítulo 2 del libro *Principia Mathematica* de Whitehead y Russell. Sin embargo, lo que nos interesa aquí no es resaltar la capacidad del programa para demostrar teoremas, sino uno de los pasos previos requeridos para su construcción: el desarrollo de un lenguaje de programación.

Desde su convencimiento sobre la capacidad simbólica de los ordenadores, Newell, Simon y Shaw empezaron a estudiar tareas concretas que un ordenador pudiera ejecutar. Analizaron el juego del ajedrez, la resolución de problemas geométricos y, por supuesto, la demostración de teoremas lógicos. En esto, observaron la necesidad de diseñar un lenguaje de programación que, pudiendo traducirse automáticamente al lenguaje máquina, fuese más adecuado para expresar procesamiento de formas complejas de información. Así, un equipo dirigido por Newell desarrolló un lenguaje de procesamiento de listas que permitió, en un

primer momento, la simulación manual del procedimiento de demostración de uno de los teoremas por Herbert Simon y su posterior implementación efectiva en un ordenador.

Son destacables también en el Teórico Lógico sus procedimientos para la resolución de problemas. Newell y Simon no sólo se interesaron por las demostraciones finales, sino también estudiaron con detalle las operaciones que el programa ejecutaba para alcanzar su objetivo. Es más, con el fin de obtener apoyo empírico para su hipótesis de paralelismo entre el procedimiento utilizado por el Teórico Lógico y el utilizado por los seres humanos, realizaron estudios que sirven, hoy día, como protocolos de validación de los programas de simulación, experimentando con su programa y comparando los protocolos de ejecución.

Así, la inteligencia artificial pone a disposición de los investigadores de la psicología cognitiva un lenguaje formal para expresar sus modelos teóricos y una metodología para poner a prueba dichos modelos.

1.3.1. Lenguajes de programación

Los ordenadores son máquinas que ejecutan instrucciones. Sin embargo, es importante aclarar que comandos como *dir* de los sistemas operativos D.O.S. están compuestos por un conjunto de instrucciones que el ordenador necesita ejecutar para, en este caso, listar en pantalla todos los ficheros existentes en el directorio actual. Estas instrucciones básicas o primitivas que los circuitos electrónicos de los ordenadores pueden reconocer y ejecutar directamente son instrucciones simples del tipo sumar dos números, comprobar si un número es cero, o mover datos de una parte de la memoria a otra. Este conjunto de instrucciones compone lo que se conoce como el lenguaje máquina y es tan elemental que hace difícil y tedioso su uso. El siguiente ejemplo servirá como ilustración de las características de los lenguajes máquina.

```
*****
;PROGRAMA ALUMNO.ASM
*****
;
;
=====
;DEFINICION DEL SEGMENTO DE DATOS
;
DATAS      SEGMENT
CLR_PANT   DB  "",13,10,"$"
CUR_MICRO  DB  "TECNOLOGIA DE COMPUTADORES",13,10,"$"
PREG_NOM   DB  "ESCRIBE TU NOMBRE ? $"
NOMBRE     DB  80 DUP (0)
TEXTO      DB  13,10,"DON"," $"
TEXTO2     DB  " ES ALUMNO DE 2º CURSO DE "
           DB  "INGENIERIA INFORMATICA",13,10,"$" ;AÑADE CR y LF
DATAS      ENDS
```

```

;
;=====
;DEFINICION DEL SEGMENTO DE PILA
;
PILA      SEGMENT STACK "STACK"
          DB  40H DUP (0)
PILA      ENDS
;
;=====
;DEFINICION DEL SEGMENTO DE CODIGO
;
CODIGO     SEGMENT
          ASSUME CS:CODIGO,DS:DATOS,SS:PILA
;PROCEDIMIENTO PRINCIPAL
;
START      PROC FAR
;INICIALIZAR REGISTROS DE SEGMENTOS
;
          MOV AX,DATOS                      ; B8 0000s
          MOV DS,AX                          ; 8E D8
;
;BORRAR PANTALLA
          MOV AH,9                          ; B4 09
          MOV DX,OFFSET CLR_PANT            ; BA 0000r
          INT 21H                           ; CD 21
;
;ESCRIBE EN PANTALLA
          MOV DX,OFFSET CUR_MICRO           ; BA 0003r
          INT 21H                           ; CD 21
;
          MOV DX,OFFSET PREG_NOM           ; BA 0020r
          INT 21H                           ; CD 21
;
;RECOGE EL NOMBRE DESDE EL TECLADO
          MOV AH,0AH                        ; B4 0A
          MOV DX,OFFSET NOMBRE              ; BA 0037r
          MOV NOMBRE[0],60                  ; C6 06 0037r 3C
          INT 21H                           ; CD 21
;
;ESCRIBE EN PANTALLA EL NOMBRE INTRODUCIDO
          MOV AH,9                          ; B4 09
          MOV DX,OFFSET TEXTO              ; BA 0087r
          INT 21H                           ; CD 21
;
          MOV BX,0                          ; BB 0000
          MOV DL,NOMBRE[BX+2]              ; 8A 97 0039r
          MOV AH,2                          ; B4 02
          INT 21H                           ; CD 21
          INC BX                             ; 43
          CMP BL,NOMBRE[1]                 ; 3A 1E 0038r
          JNE OTRO                         ; 75 F1
;
          MOV DX,OFFSET TEXTO2             ; BA 008Fr
          MOV AH,9                          ; B4 09
          INT 21H                           ; CD 21
;
;FIN DEL PROGRAMA Y VUELTA AL DOS
;
          MOV AX,4C00H                      ; B8 4C00
          INT 21H                           ; CD 21
;
START      ENDP
CODIGO     ENDS
;
;=====
;FIN DEL PROGRAMA, INDICANDO DONDE COMIENZA LA EJECUCION
;
          END START
;*****

```

El programa anterior está escrito en lenguaje ensamblador, que es una representación simbólica del lenguaje numérico de máquina. Existe por tanto una correspondencia unívoca entre las sentencias del programa en ensamblador y las instrucciones que la máquina ejecuta que, a modo ilustrativo, acompañan como comentarios a las sentencias que definen el segmento código en el programa ejemplo.

Para evitar que los programadores tenga que escribir algoritmos en un lenguaje diseñado más para máquinas que para humanos, se desarrollaron los lenguajes de alto nivel como COBOL, FORTRAN o BASIC. Son programas que traducen o interpretan un conjunto de nuevas instrucciones más adecuadas para las personas en instrucciones propias de la máquina. Como ejemplo, el siguiente programa escrito en el lenguaje QBASIC que realiza la misma tarea que el ejemplo anterior:

```
10 PRINT ""
20 PRINT "TECNOLOGIA DE COMPUTACION"
30 PRINT "ESCRIBE TU NOMBRE ";
40 INPUT a$
50 PRINT "DON " + a$ + " ES ALUMNO DE 2º CURSO DE INGENIERIA INFORMATICA"
60 PRINT ""
```

No hay duda de que estos lenguajes de alto nivel han facilitado la labor de los programadores. Sin embargo, los problemas de la inteligencia artificial demandaban herramientas metodológicas particulares, como señalaron Newell y Simon, un lenguaje que les permitiera pensar sin tropiezos en las tareas de resolución de problemas y simular con gran aproximación las clases de etapas mentales que presuntamente atraviesa un ser humano en tales circunstancias.

A tal empresa se dedicó otro de los fundadores de la inteligencia artificial, John McCarthy, dando lugar a la creación del lenguaje de programación por excelencia en el campo de la inteligencia artificial: el LISP. Consiste, básicamente, en un proceso sucesivo de evaluación de listas, esto es, encontrar un valor para ellas. Por ejemplo, veamos los valores obtenidos por la evaluación de las siguientes expresiones:

```
* (+ 1 2)
3
* (+ (+ 1 2) (+ 1 2))
6
* (+ (+ (+ 1 2) (+ 1 2)) (+ 1 2))
9
```

En estos ejemplos sencillos, que no necesitan comentarios adicionales para su comprensión, se observa una de las características fundamentales del LISP que le hace, más que un nuevo lenguaje, un nuevo paradigma de programación. A diferencia de los lenguajes secuenciales como el COBOL, FORTRAN o BASIC, los programas LISP no se definen en términos de secuencias ordenadas de instrucciones, sino como funciones con argumentos de entrada y valores de salida. Esta nueva conceptualización de la programación, estructurando las tareas a realizar por un programa en funciones que transforman información de entrada en información de salida, le confiere una propiedad fundamental de la que se deriva su potencia para la elaboración de programas de inteligencia artificial: la recursividad, esto es, la capacidad de una función de llamarse a sí misma. Por ejemplo, para calcular el número de elementos que componen una lista, podemos definir la siguiente función:

```
(DEFUN CONTAR (L)
  (DO ((LISTA L (REST LISTA))
      (CONTADOR 0 (+ 1 CONTADOR)))
    ((ENDP LISTA) CONTADOR)))
```

o, definir una función recursiva:

```
(DEFUN CONTAR (L)
  (IF (ENDP L)
      0
      (+ 1 (CONTAR (REST L)))))
```

Las diferencias entre estas dos funciones van más allá de estrategias de programación, reflejando formas distintas de análisis de problemas. En el primer caso, el problema se analiza en términos de la secuencia de cálculos y decisiones que se requieren para alcanzar la solución. En el segundo caso, el análisis del problema consiste en definir unos casos básicos y procedimientos de transformación que reducen sucesivamente el problema en subproblemas más simples hasta llegar a la solución a partir de la definición de los casos básicos.

Estas son algunas de las características técnicas del LISP que no sólo permitió superar las limitaciones de la programación procedural tradicional sino que, además, se adaptó a las particularidades que presentan los problemas de la inteligencia artificial. Así, el LISP (y sus descendientes) es el lenguaje de programación más utilizado en la inteligencia artificial, llegando a ser considerado como la matemática de la inteligencia artificial, el dialecto preciso e inequívoco en que se formulan las teorías (Boden, 1977; Gardner, 1985).

Los beneficios de esta nueva metodología de programación no fueron exclusivos para la inteligencia artificial. Los nuevos lenguajes de programación desarrollados para la informática convencional, como PASCAL o C, poseen sus propiedades fundamentales, aunque técnicamente se diferencian bastante del LISP. Tanto es así, que a partir de los años 80, el C fue sustituyendo progresivamente al LISP como principal herramienta de desarrollo de los programas de inteligencia artificial (Zaccagnini, 1994).

Otro lenguaje creado con el mismo propósito que el LISP y ofreciéndose como alternativo a él es el PROLOG. Es un lenguaje de programación basado en la lógica predicativa en el que los programas se definen como conjuntos de sentencias lógicas. Es fundamentalmente un lenguaje declarativo en el cual programar no consiste en decirle al ordenador lo que debe hacer y en qué momento sino decirle lo que es verdadero mediante declaraciones lógicas y que intente sacar conclusiones de ello. Por ejemplo:

```
padre (juan, pedro) .  
padre (jose, juan) .  
abuelo (x, y) :-  
    padre (x, z) ,  
    padre (z, y) .  
  
?- abuelo (jose, pedro) .  
yes
```

La programación en PROLOG consiste, por tanto, en declarar relaciones entre objetos, definir reglas sobre los objetos y sus relaciones, y hacer preguntas sobre los objetos y sus relaciones. La programación lógica comparte la potencia de la lógica predicativa para representar y razonar con conocimiento. Esta es una de las aportaciones de este tipo de programación a la inteligencia artificial. Sin embargo, hay contenidos relevantes del conocimiento difícilmente representables en términos de la lógica predicativa y, por tanto, difícilmente implementables con lenguajes de programación lógica. Al igual que el LISP, el PROLOG, como lenguaje de programación lógica más utilizado, sigue contando con defensores entre los programadores de la inteligencia artificial, aunque su uso se ha ido restringiendo progresivamente a un área específica.

Más recientemente ha surgido otro paradigma de programación denominado “programación orientada a objetos” siendo el lenguaje C++ su principal herramienta de implementación. Este nuevo método de programación consiste en aplicar los modelos organizados de los conceptos del mundo real al análisis de los problemas y al diseño de los programas. Así, los programas, en lugar de ser conjuntos de funciones, son colecciones

organizadas de unidades conceptuales. Cada una de ellas es una clase que describe un conjunto posiblemente infinito de objetos individuales que son definidos por los mismos atributos y la mismas operaciones. Estas clases de objetos no existen de forma aislada sino que conforman una organización jerárquica. Esto es, algunas clases comparten entre ellos un conjunto de atributos y operaciones que forman parte de su definición. Estas propiedades comunes definen una clase de orden superior y todas las clases que de ella se derivan heredarán los atributos y operaciones que la definen. Programar es, por tanto, construir jerarquías de clases de objetos, indicando el comportamiento y las características de cada uno.

La principal aportación de esta metodología a la inteligencia artificial es la de proporcionar una forma natural de construcción de modelos computacionales de sistemas conceptuales del mundo real. Los programas de inteligencia artificial necesitan contar con una representación funcional del escenario en que se define la tarea o problema a abordar. La representación del conocimiento es una constante en los problemas de inteligencia artificial. La lógica del paradigma orientado a objetos se presenta muy útil para ello (véase Rich y Knight, 1991; Cawsey, 1998).

El lenguaje de programación C++ ha heredado todas las propiedades del C, incorporando nuevas características adaptadas para la programación orientada a objetos. Estas son: el encapsulamiento, la herencia y el polimorfismo. Los atributos y las operaciones definen en su conjunto cada clase de objetos. El C++ y otros lenguajes orientados a objetos cuentan con una nueva estructura de datos, la clase, que se caracteriza por integrar en sí misma los dos componentes básicos de los programas informáticos: datos y funciones. Esta propiedad de las clases se conoce como encapsulamiento. La propiedad de herencia es la que permite crear una clase nueva como extensión de su clase base y así establecer una estructura jerárquica de las clases. Finalmente, el polimorfismo es la capacidad de las clases derivadas para redefinir las propiedades heredadas de las clases bases. Estas características de las clases están presentes en otros lenguajes de programación orientados a objetos (e.g., Smalltalk).

Esto es, una propiedad adicional del paradigma de programación orientada a objetos es la independencia entre las fases de análisis, diseño e implementación. Las limitaciones de los lenguajes de programación no deben estar presentes en las fases de análisis o de diseño. Sin embargo, cuenta con una metodología muy bien definida para estas fases. Esto, además de proporcionar a los investigadores una mayor flexibilidad en el análisis de los problemas, permite a científicos sin formación informática aplicar las herramientas de representación

definidas en esta metodología para la formulación de sus teorías. Estas herramientas conocidas como “tecnologías de modelado por objeto” se ofrecen como una alternativa especialmente interesante para la representación formal de teorías computacionales de la cognición.

Haciendo uso de la notación de una de estas tecnologías, el método **OMT**: *Object Modeling Technique* (Rumbaugh, Blaha, Premerlani, Eddy y Lorensen, 1991), presentamos una sencilla jerarquía de clases de un subconjunto de las figuras geométricas bidimensionales como ejemplo de esta metodología y de las propiedades fundamentales de la programación orientada a objetos.

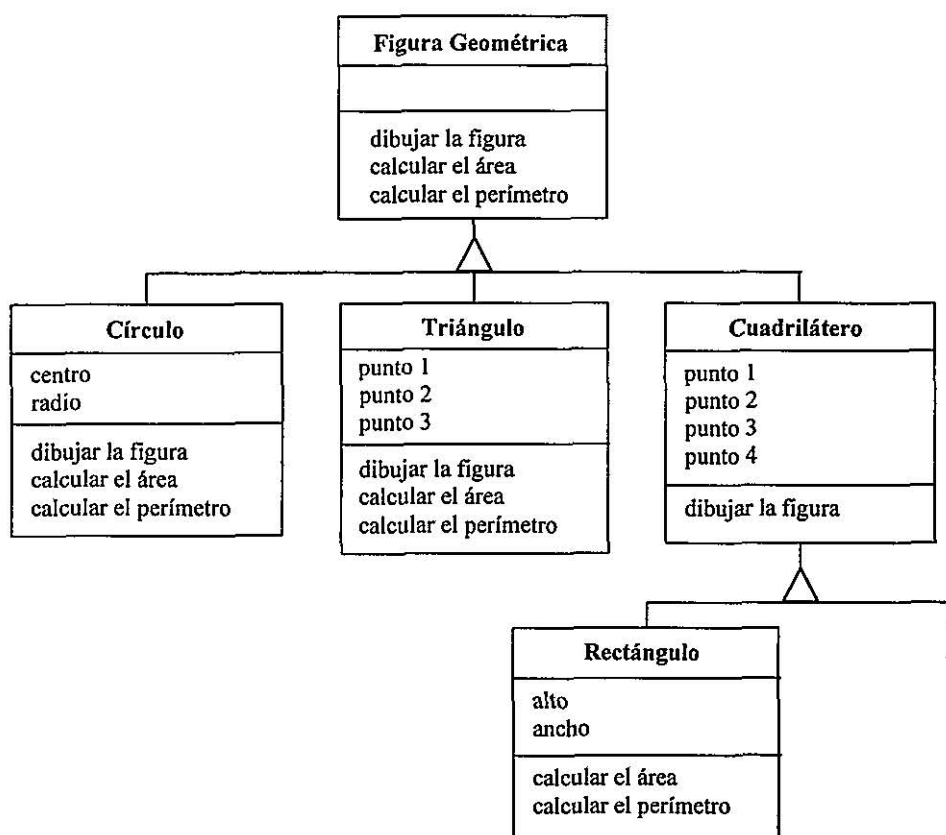


Figura 1.1.— Un ejemplo de modelo de objetos.

Comenzamos con la definición de una clase base: la clase de *Figura Geométrica*. En ella agruparemos los atributos y las operaciones comunes a todas las figuras geométricas sin distinción. Para este ejemplo proponemos las operaciones de dibujar la figura, calcular el área y el perímetro de la misma.

Evidentemente no todas las figuras se dibujan de la misma forma. Asimismo el cálculo del área y del perímetro son funciones bien diferenciadas para cada una de ellas. Sin embargo,

estas operaciones representan una de las propiedades comunes a todas las figuras geométricas bidimensionales y, por tanto, deben ser contempladas en la definición de la clase base *Figura Geométrica*, si bien, las diferencias de realización de estas operaciones forman parte, a su vez, de la definición de sus clases derivadas. Esta aparente incompatibilidad es solventada por una de las propiedades básicas del paradigma orientado a objetos: el polimorfismo. Esta característica hace referencia a la capacidad de las clases derivadas de redefinir propiedades heredadas de su clase base. Esto es lo que nos permite definir las operaciones comunes a todas las figuras geométricas aún cuando éstas serán redefinidas en cada clase en particular.

Definimos ahora tres clases derivadas de la clase *Figura Geométrica*: *Círculo*, *Triángulo* y *Cuadrilátero*. Todas ellas comparten las propiedades presentes en la clase base por la propiedad de herencia y se caracterizan por las propiedades particulares comunes a todas las figuras que representan. Así, la clase *Círculo*, además de la redefinición de las operaciones heredadas, se define por atributos propios como son el centro y el radio. De igual modo, las clases *Triángulo* y *Cuadrilátero* son definidas por 3 y 4 vértices respectivamente. En la definición de estas clases observamos otra de las propiedades distintivas de la metodología que estamos analizando, la integración de datos y funciones: el encapsulamiento.

Seguimos desarrollando la jerarquía en una de las clases derivadas de la clase *Figura Geométrica*, la clase *Cuadrilátero*. Es conocido que los cuadriláteros, a pesar de compartir atributos en común (tener 4 lados o 4 vértices) no son todos iguales. Así, por ejemplo, podemos diferenciar aquellos que tienen sus lados paralelos de los que no los tienen. Incluso el cálculo del área y del perímetro es diferente para distintos tipos de cuadriláteros, por ejemplo, un rectángulo y un trapecio. Esto es un indicador de que de ella pueden ser derivadas otras clases.

En el ejemplo hemos definido la clase *Rectángulo* como una de las posibles clases derivadas de la clase *Cuadrilátero* para explicitar un caso claro de herencia de atributos y operaciones. Para la definición de esta nueva clase, nos hemos servido de los atributos alto y ancho y la redefinición de las operaciones de cálculo de área y perímetro. Sin embargo, en la definición de la clase *Rectángulo* están presentes todas las propiedades de la clase *Cuadrilátero*. Así, la clase *Rectángulo* integra en su definición los cuatro vértices y la operación de dibujar heredadas de su clase superior y comunes a otras clases derivadas de dicha clase.

Con este ejemplo no sólo hemos procurado ilustrar las tres propiedades fundamentales del paradigma orientado a objetos, sino poner de manifiesto la ausencia de conocimiento de

programación informática en el uso de esta herramienta. Como se ha podido observar, el desarrollo del ejemplo está fundamentado en el conocimiento de la geometría y la notación de representación de una tecnología particular.

```
class Circulo : public FiguraGeometrica {  
    protected:  
        int centro;  
        int radio;  
    public:  
        void DibujarFigura ();  
        float CalcularArea ();  
        float CalcularPerimetro ();  
};
```

Esta implementación de la clase *Circulo* en C++ no es relevante ni interviene en el proceso de análisis y diseño del modelado orientado a objetos.

Hasta este momento nos hemos referido únicamente a paradigmas y lenguajes de programación de propósito general. Existen también otras herramientas para la construcción de aplicaciones de inteligencia artificial que han sido desarrolladas específicamente para abordar determinados tipos de problemas. Estas herramientas se caracterizan por llevar implementados procedimientos generales válidos para la mayoría de los sistemas que tuviesen una representación similar del problema. Los entornos de desarrollo de sistemas basados en reglas como EMYCIN, PCPLUS o M4 son ejemplos de este tipo de herramientas. Los sistemas basados en reglas están compuestos por un conjunto de reglas que establecen relaciones condicionales entre hechos o acciones, un conjunto de hechos y un programa de control de aplicación de las reglas según los hechos. Las reglas son particulares del dominio y los hechos son específicos para cada caso, pero tanto la estructura de las reglas como el proceso de inferencia son, en general, comunes para los sistemas basados en reglas. Éstos son los componentes implementados en estos entornos como editor de reglas y motor de inferencia. Así, los programadores de este tipo de sistemas no tendrán que crear estructuras de datos para representar las reglas ni construir funciones para establecer la aplicación de reglas, únicamente tendrán que definir su ámbito de conocimiento en términos de "si ..., entonces ...". Como ejemplo, vamos a ver algunas secuencias del comportamiento de un sistema de estas características:

Regla 1 de 20 :

Si

EXISTE LA FELICIDAD

-Y-

TE SIENTES PLENAMENTE FELIZ

Entonces

DECISION

-Y-

ERES FELIZ (PORQUE TU LO DICES)

Regla 2 de 20 :

Si

HAY ALGO QUE DESEES LOGRAR Y NO PUEDES POR FALTA DE MEDIOS

-Y-

PODRIAS LOGRAR LO QUE DESEAS CON UN POCO MAS DE DINERO

Entonces

DECISION

-Y-

TIENES PROBLEMAS DE DINERO

Regla 3 de 20 :

Si

TE PREOCUPA ALGUIEN QUE TE DA MAS AMOR DEL QUE TU ERES CAPAZ DE DEVOLVER

Entonces

TIENES TRISTEZA PORQUE NO TE QUIEREN COMO TU QUISIERAS

...

Regla 18 de 20 :

Si

~SABES VERDADERAMENTE COMO TE SIENTES

-Y-

~TE SIENTES PLENAMENTE FELIZ

-Y-

~TE SIENTES MUY INFELIZ

Entonces

DECISION

-Y-

ERES NORMALMENTE FELIZ

Regla 19 de 20 :

Si

SABES VERDADERAMENTE COMO TE SIENTES

-Y-

~TIENES TRISTEZA PORQUE NO TE QUIEREN COMO TU QUISIERAS

-Y-

~TIENES PROBLEMAS DE SALUD

-Y-

~TIENES PROBLEMAS DE DINERO

Entonces

DECISION

-Y-

ERES BASTANTE FELIZ (AUNQUE NO QUIERAS RECONOCERLO)

Regla 20 de 20 :

Si

~EXISTE LA FELICIDAD

Entonces

DECISION

-Y-

ENTONCES NO PUEDES SER FELIZ PORQUE ES IMPOSIBLE

Las siguientes dos secuencias han sido registradas en la ejecución de la implementación de estas reglas en el entorno de desarrollo PSP:

Es cierto que ... :	
EXISTE LA FELICIDAD	SI
Es cierto que ... :	
TE SIENTES PLENAMENTE FELIZ	NO
Es cierto que ... :	
TIENES PROBLEMAS DE DINERO	NO
Es cierto que ... :	
TIENES TRISTEZA PORQUE NO TE QUIEREN COMO TU QUISIERAS	SI
Es cierto que ... :	
TIENES PROBLEMAS DE SALUD	NO
DECISION	
TE SIENTES MUY INFELIZ (PERO PODRIAS ESTAR PEOR)	
Es cierto que ... :	
EXISTE LA FELICIDAD	SI
Es cierto que ... :	
TE SIENTES PLENAMENTE FELIZ	LO
IGNORO	
Es cierto que ... :	
TIENES PROBLEMAS DE DINERO	NO
Es cierto que ... :	
TIENES TRISTEZA PORQUE NO TE QUIEREN COMO TU QUISIERAS	LO
IGNORO	
Es cierto que ... :	
TIENES PROBLEMAS DE SALUD	NO
Es cierto que ... :	
TE SIENTES MUY INFELIZ	LO
IGNORO	
Es cierto que ... :	
TE PREOCUPA ALGUIEN QUE TE DA MAS AMOR DEL QUE TU ERES CAPAZ DE DEVOLVER	NO
Es cierto que ... :	
TE PREOCUPA QUE NO TE QUIEREN COMO TU QUISIERAS	NO
Es cierto que ... :	
MAS O MENOS ESTAS HACIENDO LO QUE DESEAS	SI
Es cierto que ... :	
SABES VERDADERAMENTE COMO TE SIENTES	NO
DECISION	
ESTAS NORMAL (O SEA, NI FU NI FA)	

La psicología cuenta con estas herramientas para desarrollar y expresar sus teorías. La ventaja adicional de este tipo de formalización es la capacidad para implementar teorías científicas en los ordenadores. Expresar las teorías en forma de programas de ordenador y comprobar que funcionan al ser ejecutados por una máquina nos ofrece garantías sobre su corrección, indicando si están bien o mal construidas. Sin embargo, esto por sí mismo, no indica nada acerca de su veracidad. Sobre esta cuestión nos referiremos a continuación.

1.3.2. Simulación

Así como no todos los programas de inteligencia artificial pueden ser considerados como teorías psicológicas, no todas las teorías así formalizadas e implementadas son válidas para explicar el comportamiento humano. Sin duda, con ello tenemos una teoría explicativa del comportamiento del ordenador, en el sentido de que la salida producida depende causalmente del programa. Su validez como teoría explicativa del comportamiento humano dependerá de su capacidad para simular los mecanismos subyacentes a dicho comportamiento. Para comprender

adecuadamente la simulación por ordenador como explicación de la conducta debemos aclarar previamente el concepto de simulación (García-Albea, 1981).

Lo primero que debemos tener en cuenta es que si X simula Y entonces X no es Y. La relación entre un sistema simulador con el sistema simulado no es una relación de identidad sino de semejanza y como tal no es una cuestión de todo o nada. Esto es, la semejanza entre dos sistemas puede ser de mayor o menor grado según la cantidad de elementos comunes compartidos por ambos. Por esto, es imprescindible especificar los aspectos sobre los cuales se establece tal relación de semejanza.

Por tanto, la simulación se realiza sobre algunas propiedades de los aspectos concretos de una determinada parcela de la realidad. Su definición es fundamental para la evaluación de los resultados de la simulación. Por ejemplo, la cámara fotográfica puede ser considerada una buena simulación del ojo como aparato óptico pero no del ojo como aparato sensorial. En este caso, la lectora de CD-ROM puede considerarse una simulación más adecuada.

Ahora bien, ¿qué condiciones deben cumplir los programas de simulación para ser considerados auténticas teorías psicológicas? Éstas no son ni más ni menos que las que se le exige a cualquier teoría para su validación (Fodor, 1968; García-Albea, 1981).

La primera condición hace referencia a aspectos de la conducta observable. La salida proporcionada por la máquina tendría que ser indistinguible del comportamiento del organismo en aquellos elementos que describen nuestro objeto de estudio, es decir, en aquellos aspectos relevantes para su explicación. Esto es, eliminar de la comparación las variables que, no siendo relevantes para nuestro estudio, sí influyen en su valoración. No hay dudas de que si no se establecen algunas restricciones, como que los jueces no puedan ver los sistemas evaluados o que las respuestas sean recibidas en el mismo formato, ningún sistema artificial, ni siquiera C-3PO, podría pasar la prueba de Turing. Esto no es una cuestión distinta al control de variables extrañas de los estudios experimentales. Así, la valoración de los resultados de la simulación, al igual que la interpretación de los datos experimentales, no debe ser independiente de la teoría psicológica a la que trata de modelar.

Sin embargo, la explicación psicológica trasciende más allá de los elementos puramente observables. El conductismo tuvo que ampliar su marco de referencia para dar cuenta de las conductas potenciales. Por ello mismo, la simulación debe contar con un repertorio de

conductas que, además de producir las conductas actuales, nos permitan predecir con exactitud el resultado si las condiciones fueran otras distintas de las que se observan.

Ahora bien, la mayoría de los programas de inteligencia artificial cumplen estos dos requisitos que son imprescindibles para la valoración del éxito de la mayoría de los proyectos de desarrollo. El famoso jugador artificial de ajedrez, Deep-Blue, puede servir como ejemplo. Las conductas observadas en sus partidas con Kasparov, con las restricciones pertinentes, son indistinguibles de las de un jugador humano. Resulta casi evidente inferir, según su comportamiento a lo largo de la competición, que tales conductas forman parte de un repertorio más general. Pero no por ello es considerado una teoría explicativa de la conducta de jugar ajedrez. Uno de los aspectos más llamativos de su comportamiento es que, pese a su potencia de cálculo y de búsqueda y su capacidad de memoria, muy superior al humano, sus tiempos de decisión no sólo no son significativamente menores sino que en muchos casos resultan sorprendentemente lentos. Este es un indicador de que hay propiedades causales de la conducta que no están contempladas en el Deep-Blue. Una explicación psicológica de la conducta debe dar cuenta de ellas. Esta es la tercera condición requerida para que la simulación sirva como una auténtica teoría psicológica.

Los fenómenos mentales, en su papel causal de la conducta inteligente, hacen imprescindible su referencia en la explicación psicológica. Por tanto, la simulación por ordenador no sólo debe hacer lo mismo sino que también debe hacerlo de la misma manera que el organismo simulado, replicando así los procesos y las representaciones subyacentes a la producción de la conducta.

La validez de la simulación como teoría psicológica radica en la correspondencia entre las operaciones del programa de simulación y los procesos efectuados por el sujeto. Esta correspondencia se establece en los siguientes aspectos. Por un lado, la simulación debe presentar una relación equivalente a la relación en el organismo entre los procesos cognitivos y la conducta, esto quiere decir que entre el programa de simulación y su salida debe establecerse una relación de dependencia causal. Por otro lado, entendiendo que una de las tareas de una teoría psicológica es identificar los procesos que dan lugar a distintos tipos de conducta, sería imprescindible que, a un nivel de descripción adecuado, a cada operación del programa de simulación le corresponda un proceso del organismo (García-Albea, 1981).

La simulación no sólo se ofrece como una posible teoría psicológica explicativa de la conducta humana, sino que, al mismo tiempo, resulta un modelo de la misma. Esto es, los programas de simulación son expresiones teóricas que tienen un comportamiento propio. Esta propiedad nos permite observar el funcionamiento real de la teoría y así ponerla a prueba y experimentar con ella (Zaccagnini, 1994). Con esto, los modelos teóricos tienen la capacidad de generar hipótesis precisas con predicciones bien definidas, lo que nos permite interpretar de forma unívoca los posibles resultados experimentales y sus implicaciones teóricas.

La inteligencia artificial y la psicología cognitiva comparten un mismo objeto de estudio: las funciones cognitivas. El desarrollo de la inteligencia artificial como disciplina científica no ha sido, en absoluto, independiente de otras disciplinas como la psicología o la lingüística. Por tanto, no es casual observar la utilidad de sus herramientas metodológicas para la expresión teórica y construcción de modelos psicológicos. Sin embargo, el uso de estas herramientas debe hacerse respetando sus propias restricciones.

En psicología es frecuente hacer uso de los diagramas de flujo como expresión gráfica de sus teorías. Los diagramas de flujo en la programación convencional son esquemas simbólicos de determinados procesos de tratamiento de datos en términos de los elementos implicados en el sistema. En ellos se detallan los pasos a realizar en un proceso y las relaciones entre los diferentes pasos y los componentes del sistema utilizados en cada uno de ellos. Sin embargo, la mayoría de los modelos psicológicos representados en diagramas de flujo no cumplen uno de sus requisitos básicos, y es que cada componente del diagrama debe describirse en términos de operaciones elementales bien definidas. Este no es otro requisito que exigir que los elementos teóricos que componen los modelos psicológicos estén explicados o bien en la propia teoría o bien en otras afines.

Podríamos resumir en tres puntos las implicaciones de la inteligencia artificial en la construcción teórica de la psicología (Adarraga, 1994):

- ♦ La formalización de teorías psicológicas en lenguajes de programación permite, por medio de la nueva tecnología, la instanciación objetiva de sus conceptos teóricos, más allá de simples palabras.

- ♦ Los programas de simulación como modelo sirven para poner a prueba las teorías experimentando con los fenómenos descritos en tales teorías.
- ♦ El ordenador nos permite observar paso a paso los procesos que se ejecutan, pudiendo identificar, así, las operaciones y estructuras de datos implicados en cada momento. De este modo, los programas de simulación constituyen una fuente de primer orden para la elaboración teórica propiamente dicha en psicología.

Una vez expuestos con detalle los aspectos principales del marco teórico y metodológico en el que nos encuadramos, vamos a analizar nuestro objeto de estudio: los procesos de producción del lenguaje.

Capítulo 2: Procesos de producción del lenguaje

2.1. Introducción

Desde el enfoque del procesamiento de la información, la psicolingüística define el lenguaje como componente funcional y comportamiento derivado de un organismo activo, describiendo la actividad lingüística en términos de procesos y representaciones mentales. Esto es, la actividad lingüística es producto de la intervención de un conjunto de operaciones de computación específicos sobre las representaciones, tanto del conocimiento lingüístico como de otros sistemas de conocimiento. Así, el objetivo de los psicolingüistas interesados en el estudio de las actividades de producción y comprensión del lenguaje es establecer los mecanismos funcionales que subyacen al comportamiento lingüístico, determinando los procesos psicológicos y representaciones mentales involucrados en el procesamiento lingüístico.

La psicolingüística, como psicología científica y disciplina empírica, elabora sus modelos teóricos atendiendo a los resultados de los estudios empíricos. Para ello, cuenta con los siguientes métodos de investigación: la observación, la experimentación y la simulación.

El método de observación consiste en la selección, registro objetivo y codificación de parámetros lingüísticos y/o extralingüísticos de las conductas lingüísticas. Los datos, así obtenidos, nos permiten describir y catalogar distintas formas de actuación lingüística y establecer ciertas relaciones hipotéticas entre variables o fenómenos asociados a su realización desde las regularidades observadas.

Sin embargo, para la contrastación empírica de hipótesis sobre relaciones causales o funcionales entre eventos es imprescindible el uso de la experimentación, diseñando situaciones artificiales y totalmente controladas de observación. Esto es, observar y medir los cambios producidos en las variables a explicar (variables dependientes) por la manipulación de forma reversible de ciertas variables hipotéticamente explicativas (variables independientes), controlando variables cuyo efecto es considerado irrelevante para los propósitos teóricos del experimento (variables extrañas).

Una tercera opción metodológica es la simulación. Se define, fundamentalmente, como la elaboración de programas de ordenador que, siguiendo las especificaciones de la teoría, sirven para la contrastación de la misma. Por tanto, en la investigación psicolingüística y, en general, en la psicología cognitiva del procesamiento de la información, la simulación consiste, básicamente, en experimentar fenómenos reales con modelos.

Es importante considerar, a la hora de aplicar estas herramientas metodológicas al estudio de los procesos psicolingüísticos, las peculiaridades derivadas de la naturaleza propia del lenguaje. En primera instancia, el carácter creativo del lenguaje da lugar, de por sí, a amplias diferencias inter e intra individuales. Por otra parte, su naturaleza comunicativa hace la actividad lingüística especialmente sensible a variables contextuales así como funcionalmente dependiente de la conducta de otros individuos. Estas propiedades del lenguaje dificultan tanto la interpretación de los datos observados como el control de condiciones contextuales, sin interferir en la propia actividad lingüística. Es más, su complejidad cognitiva dificulta la elaboración de hipótesis sobre los mecanismos implicados en la actividad lingüística de forma suficientemente precisa, que permita establecer predicciones empíricas bien definidas para una interpretación no ambigua de los datos.

Estas dificultades metodológicas son especialmente evidentes en el estudio de la producción, donde resulta casi imposible manipular de modo reversible algunos aspectos iniciales del procesamiento, como los contenidos mentales y las variables motivacionales del sujeto. Esto se observa en el desfase de la investigación en producción respecto a la comprensión y en las estrategias metodológicas utilizadas para su estudio, fundamentalmente medidas observacionales (Fodor, Bever y Garrett, 1974; Butterworth, 1980a; Levelt, 1989; Belinchón, Rivière e Igoa, 1992).

No obstante, en la medida en que se están desarrollando herramientas metodológicas específicas para el estudio de la producción verbal, los estudiosos de la producción del lenguaje cuentan con un considerable cúmulo de datos empíricos en base a los cuales formulan sus explicaciones teóricas acerca de la naturaleza de las representaciones y los procesos implicados. Con todo, la psicolingüística cuenta actualmente con un esquema general de descripción funcional de los componentes básicos que configuran el sistema de procesamiento lingüístico (Levelt, 1989) y propuestas sólidas de modelos de codificación gramatical en producción (Garrett, 1980, 1988; Bock y Levelt, 1994). A ello nos referiremos, con detalle, inmediatamente

después de una breve presentación de las principales estrategias metodológicas utilizadas en la investigación psicolingüística de la producción verbal.

2.2. Principales fuentes de datos empíricos

Una de las fuentes de datos observacionales que más ha contribuido a la comprensión de los procesos de producción del lenguaje son los errores espontáneos del habla. Más de una vez hemos sido objeto de risas por emitir frases como “esa carta es una *protuesta*” o “aquí estamos como *lata*s en *sardina*”. Sin embargo, el análisis sistemático de estos errores espontáneos del habla muestra regularidades que nos permiten inferir algunas propiedades de los mecanismos y representaciones implicados (Fromkin, 1973, 1988; Garrett, 1975, 1980, 1988, 1991; de Viso, 1990).

En primer lugar, como podemos observar en el primer ejemplo presentado, los errores pueden ser producidos por la presencia simultánea de planes lingüísticos alternativos. En el ejemplo, la fusión entre dos alternativas léxicas, “propuesta” y “protesta”, dio lugar al error “*protuesta*”. Siguiendo este criterio, Butterworth (1982) establece tres tipos de errores espontáneos del habla: errores por planes conceptuales en competencia, errores por planes lingüísticos alternativos y errores internos al plan lingüístico. El segundo ejemplo corresponde a este último tipo de errores.

En el segundo ejemplo, el error se produce por el intercambio de posición en la frase de los morfemas raíces “*lata*” y “*sardina*”. Un primer aspecto a destacar de este ejemplo es el fenómeno de anclaje. Esto es, errores donde hay movimiento de morfemas raíces en los que los sufijos permanecen en sus posiciones en la frase. Otra regularidad observada en errores de este tipo es la correspondencia de los elementos implicados en cuanto a su categoría lingüística. La frase “hemos hecho el *chocolate* relleno de chocolate” es otro ejemplo en el que el elemento erróneo “*chocolate*” y el elemento pretendido “*bizcocho*” son unidades lingüísticas de la misma categoría, en este caso, palabras de clase abierta o de contenido. En este último ejemplo, el error se produce por la anticipación de la palabra “*chocolate*” que sustituye a la palabra “*bizcocho*” y que a diferencia de los errores de intercambio no sustituye, a su vez, a la palabra “*chocolate*”.

Por tanto, el análisis y la clasificación de los errores producidos en la formulación lingüística de un único plan conceptual pueden ser realizados siguiendo dos criterios

independientes: la unidad afectada por el error y la naturaleza del error (Garrett, 1980; del Viso, 1990).

A. Unidad afectada por el error.

- A.1. Unidades subléxicas: fonemas o grupos de fonemas que no constituyen una unidad morfológica.
- A.2. Unidades léxicas: morfemas ligados -raíces, afijos- o palabras completas.
- A.3. Unidades supraléxicas: unidades mayores que la palabra hasta un sintagma completo.

B. Naturaleza del error.

- B.1. Contextuales: el error se produce por la aparición de un elemento de la cadena en una posición incorrecta.
 - B.1.1. Anticipación: el elemento error, además de su posición original, aparece en una posición anterior.
 - B.1.2. Perseveración: cuando el elemento error se repite en una posición posterior a la original.
 - B.1.3. Intercambio: se produce una trasposición de dos elementos, donde cada uno ocupa la posición original del otro.
 - B.1.4. Desplazamiento: se caracteriza por el movimiento de un elemento de la cadena que dejando vacío su posición original pasa o ocupar otra posición en la cadena.
- B.2. No contextuales: el elemento error no se deriva de otros elementos de su contexto lingüístico inmediato.
 - B.2.1. Sustitución: el elemento pretendido es reemplazado por el elemento error.
 - B.2.2. Fusión: el elemento error resulta de la mezcla de dos posibles elementos pretendidos.
 - B.2.3. Omisión: el error se produce por la desaparición de un elemento de la cadena.
 - B.2.4. Adición: se produce la intromisión del elemento error en la cadena.

Cuadro 2.1.— Dimensiones básicas empleadas en el análisis y clasificación de los errores espontáneos del habla.

La utilidad del análisis de los errores espontáneos del habla estriba en la consistencia de las regularidades observadas. El fenómeno de anclaje y la correspondencia categorial de las unidades implicadas en el error son algunas de ellas. Así, por ejemplo, el fenómeno de anclaje, en el que el error afecta únicamente los morfemas raíces dejando intactos los sufijos, ha sido considerado como indicador empírico de la realidad psicológica de unidades lingüísticas subléxicas como los morfemas, distinguiendo, a su vez, los morfemas raíces de los afijos. La correspondencia categorial descrita para este fenómeno ha sido observada con otras unidades lingüísticas implicadas tanto en los errores contextuales como los errores no contextuales, aportando, así, datos empíricos para el establecimiento de las distintas unidades de planificación que participan en los distintos momentos del proceso de codificación lingüística.

La realidad psicológica de las estructuras lingüísticas como unidades de planificación del proceso de codificación lingüística ha sido contrastada, a su vez, por el análisis de los patrones

temporales del habla. Esto es, en el habla espontánea, pese a su aparente continuidad, se observan periodos de silencio o pausas vacías y otros fenómenos de vacilación -pausas llenas- como titubeos (“eee”, “mmm”, etc.) y “coletillas” (“o sea”, “pues”, “entonces”, etc.). Basándose en una lógica similar a los estudios con tiempos de reacción, estas interrupciones son consideradas como indicadores de los puntos del discurso con mayor carga de procesamiento cognitivo, presuntamente dedicado a la planificación. Así, el análisis de la duración y el patrón de distribución de las pausas (vacías y/o llenas) han sido utilizados, fundamentalmente, para esclarecer cuestiones relacionadas con la naturaleza de las unidades empleadas en la planificación de mensajes verbales (Maclay y Osgood, 1959; Goldman-Eisler, 1961, 1967, 1972; Henderson, Goldman-Eisler y Skarbeck, 1965; Butterworth, 1980b; Ford, 1982; Holmes, 1988).

A nivel de oraciones individuales, se han observado que la frecuencia y la duración de las pausas inter-clausales son mayores que las intra-clausales (Hawkins, 1971; Goldman-Eisler, 1972; Butterworth, 1980b). De éstas últimas, hipotéticamente asociadas a procesos de decisión léxica, se observó una mayor proporción de las pausas que preceden a palabras de contenido frente a las palabras de función (Maclay y Osgood, 1959). Asimismo, cláusulas de diferentes tipos sintácticos parecen diferir en su dificultad de producción al observar diferencias en la frecuencia y duración de las pausas que les preceden (Goldman-Eisler, 1972; Ford, 1982; Holmes, 1988).

Estos datos han sido utilizados para establecer la cláusula como unidad de planificación gramatical y apoyar la distinción psicológica entre las palabras de contenido y de función. A su vez, las diferencias observadas entre las pausas inter e intra-clausales, tanto en su frecuencia como en su duración, sugieren un mayor esfuerzo cognitivo en los procesos de decisión gramatical que los de selección léxica. Sin embargo, no se debe obviar la opacidad de estas medidas a la hora de valorar las inferencias realizadas.

Asimismo, se han realizado diversos estudios en los que se analizaron la distribución de las pausas en secuencias multioracionales (Henderson, Goldman-Eisler y Skarbeck, 1965, 1966; Butterworth, 1975, 1980b). En estos estudios, las pausas no son consideradas de forma individual, sino en su proporción global. Así, en el habla espontánea, se observaron patrones cíclicos de alternancia entre periodos vacilantes con alta concentración de pausas relativamente largas y periodos fluidos, fundamentalmente de fonación. Este resultado y la correspondencia significativa encontrada entre los ciclos y las “ideas” son considerados indicadores de la

implicación, en los procesos de producción verbal, de unidades conceptuales de nivel supraoracional como unidades de planificación, fundamentalmente semánticas.

Estas medidas observacionales, los errores y las pausas del habla espontánea, constituyen las principales fuentes de datos en la investigación psicolingüística de los procesos de producción del lenguaje. Sin embargo, pese a las dificultades ya comentadas, se han desarrollado diversas tareas experimentales que han proporcionado datos empíricos acerca de la naturaleza de los mecanismos implicados en la producción del lenguaje.

Uno de los paradigmas experimentales más utilizados en los estudios psicolingüísticos es el "priming". Consiste, básicamente, en la inducción artificial de una activación diferencial de determinadas representaciones. Esto se consigue, generalmente, mediante la presentación de un elemento relacionado con la representación "target". Así, por ejemplo, la presentación de una palabra inductora (o "prime") asociada con el nombre de la entidad "objetivo" (o "target") representada en un dibujo, constituye el procedimiento de "priming" léxico utilizado por algunos investigadores para estudiar el papel que los procesos de selección léxica desempeñan en la codificación sintáctica (Bock, 1986, 1987; Igoa, 1996). En estos trabajos se observó que, en una tarea de descripción de dibujo, la presentación de una palabra asociada semánticamente con uno de los objetos del dibujo aumentaba la probabilidad de que dicho objeto fuese designado como sujeto gramatical. Esto es, la activación semántica del léxico, por el efecto del "priming" léxico, produce la facilitación de la selección de ciertas estructuras sintácticas de la oración. Este resultado, además de apoyar la participación del léxico en el proceso de selección de marcos sintácticos, favorece la hipótesis de la disociación entre la representación de la forma y del significado en el léxico mental, ya que tal efecto de facilitación observada con el priming léxico de tipo semántico no se produce cuando la relación entre la palabra inductora y la inducida es fonológica.

Los efectos de disociación en la activación de *lemmas* y *lexemas*, así como su estricto ordenación temporal, son observadas, a su vez, en experimentos de interferencia palabra-dibujo (Glaser y Döngelhoff, 1984; Schriefers, Meyer y Levelt, 1990). Este procedimiento, utilizado para el estudio de los procesos de acceso al léxico, consiste en la presentación concurrente de una palabra y un dibujo de un objeto que los sujetos deben denominar. Manipulando, por un lado, el tipo de relación (semántica, formal o neutra) entre las palabras distractora y "target" (nombre del dibujo) y, por otro, el intervalo temporal entre la presentación de la palabra y del dibujo (asincronía estimular o SOA), el procedimiento de interferencia permite la comprobación

tanto de la naturaleza de los efectos producidos como su grado de disociación temporal. Así, se han observado dos tipos de efectos: (1) la interferencia semántica con valores negativos de asincronía estimular, esto es, cuando la presentación de una palabra semánticamente relacionada precede a la presentación del dibujo (-150 milisegundos en Schriefers *et al.*, 1990); y (2) la facilitación fonológica con SOAs de 0 y +150 milisegundos. Estos resultados, la disociación entre los efectos observados y su ordenación temporal, han contribuido a la consolidación de la propuesta de una teoría de dos fases del sistema de acceso al léxico en la producción.

Por último, debemos hacer una breve referencia a la metodología de la simulación. La psicolingüística, al igual que otras ramas de la psicología cognitiva del procesamiento de la información, ha contado con ella como una herramienta metodológica adicional. Sin embargo, en muchos casos, el papel de la simulación no iba más allá de una opción de gran interés a considerar. Actualmente, con la elaboración de modelos computacionales, la simulación está adquiriendo una mayor importancia en la investigación psicolingüística. Como ejemplo de la aplicación de esta metodología al estudio de los procesos implicados en la producción verbal presentamos el trabajo de Roelofs (1992) sobre el acceso al léxico.

Un requisito imprescindible para llevar a cabo la simulación por ordenador es contar con una formalización computacional del proceso en cuestión. A modo ilustrativo, nos referiremos a dos de los aspectos básicos del modelo propuesto en este trabajo: la estructura representacional del conocimiento léxico y el mecanismo de la selección léxica.

El léxico mental es representado en una red semántica de tres niveles. En el nivel conceptual, el significado de las palabras es representado por los nodos y sus conexiones. Cada nodo de este nivel representa un concepto particular y las conexiones entre ellos establecen las relaciones entre los distintos conceptos. Un segundo nivel, el sintáctico, contiene las propiedades sintácticas de las palabras. Los nodos de este nivel, *lemmas*, cuentan con un conjunto de conexiones. Por un lado están las conexiones con los nodos de los otros dos niveles, y por otro, las conexiones a las distintas propiedades sintácticas como la categoría gramatical o el género. Por último, las propiedades morfológicas y fonológicas están representadas en un tercer nivel, el de las formas léxicas o *lexemas*.

El proceso de selección léxica es definido, aquí, como la recuperación de la entrada léxica más activada en un proceso de activación por propagación. Esto es, en una representación en red semántica del conocimiento léxico, la activación de nodos conceptuales se propaga hacia nodos

lemmas de entre los cuales se recupera la información léxica asociada al nodo de mayor activación. Este valor de activación es un número real positivo obtenido con la siguiente ecuación:

$$a(m, t + \Delta t) = a(m, t)(1 - d) + \sum_{n \in N} w(n, m)a(n, t)$$

donde, $a(m, t)$ es el nivel de activación del nodo m en el momento t , d es la tasa de disminución progresiva asociada al nodo m ($0 < d < 1$), y Δt es la duración entre dos momentos temporales t_1 y t_2 . El segundo término de la ecuación indica el total de activación recibida por el nodo m entre t y $t + \Delta t$, donde $a(n, t)$ es el nivel de activación del nodo n en el momento t y $w(n, m)$ el peso de la conexión entre los nodos n y m , siendo N el conjunto de nodos n con conexión directa con el nodo m .

Asimismo, la simulación exige especificaciones adicionales. En primer lugar, la definición precisa de las funciones de entrada al sistema, adaptadas al procedimiento de simulación a realizar. Por ejemplo, para la simulación del proceso de recuperación del *lemma* en una tarea de denominación de dibujos con priming léxico, Roelofs definió funciones adicionales de activación de nodos conceptuales y nodos *lemmas* que simula, respectivamente, el efecto de la presentación de un dibujo y una palabra. A su vez, resulta imprescindible establecer la función de salida adecuada para el posterior análisis de ejecución. En este caso particular, se definió una ecuación para la estimación del tiempo de latencia en la recuperación del *lemma*, un índice de ejecución del sistema de simulación paralelo al empleado en este tipo de estudios con sujetos reales.

Mediante la simulación se obtienen indicadores objetivos del comportamiento del modelo teórico. Esta es la principal aportación del método de simulación, proveer a los modelos computacionales de un instrumento para la generación empírica de hipótesis. Así, por ejemplo, Roelofs observó en sus simulaciones un incremento en los tiempos de latencia estimados cuando se presentaban palabras distractoras en una tarea de denominación de dibujo, siendo, este incremento, mayor en los casos en que existía una relación semántica entre la palabra distractora y la palabra asociada al dibujo. Estas predicciones, como toda hipótesis derivada desde la teoría, deben ser contrastadas con datos obtenidos en investigaciones con sujetos reales. En el caso particular del efecto de interferencia semántica, tal como hemos comentado anteriormente, el modelo cuenta con datos previamente obtenidos en estudios experimentales realizados con sujetos reales que describen este mismo fenómeno (Glaser y Döngelhoff, 1984).

Con esto concluimos este apartado, dedicado a cuestiones metodológicas, para pasar a la exposición de las principales implicaciones teóricas derivadas de los resultados obtenidos de una extensa investigación empírica.

2.3. Caracterización funcional del sistema de producción del lenguaje

Desde la perspectiva psicolingüística, la producción verbal se describe desde la concepción del sujeto hablante como un procesador de información que realiza un conjunto de operaciones de transformación de representaciones. Esto es, operaciones que transforman una representación inicial de ciertos contenidos comunicativos en una secuencia de sonidos que puede ser percibida, analizada e interpretada por otras personas.

En el esquema general propuesto por Levelt (1989) están especificados los componentes básicos que configuran el sistema de producción del lenguaje. Este autor distingue tres componentes principales: conceptualizador, formulador y articulador, describiendo, cada uno de ellos, una función compleja que opera sobre la información de entrada para elaborar una representación de salida que puede constituirse, a su vez, como la entrada del siguiente componente.

El conceptualizador es el componente encargado de construir los llamados mensajes preverbales a partir de una intención comunicativa. Para llevar a cabo esta tarea intervienen dos grupos de procesos: los procesos de macroplanificación y los procesos de microplanificación. Los procesos de macroplanificación son los primeros en ejecutarse. Su función es la de analizar la intención comunicativa que pone en marcha el sistema de producción y especificar las unidades de información a transmitir y su modo de expresión. Estos mecanismos de selección de contenidos semánticos y pragmáticos operan a tres niveles del discurso: el discurso global, los grupos de enunciados que lo componen y cada uno de los enunciados individuales. Las especificaciones pragmáticas y las relaciones predicativas entre las unidades conceptuales seleccionadas de cada enunciado individual, junto con su orden relativo dentro del discurso, constituyen la representación de entrada del siguiente subcomponente del conceptualizador.

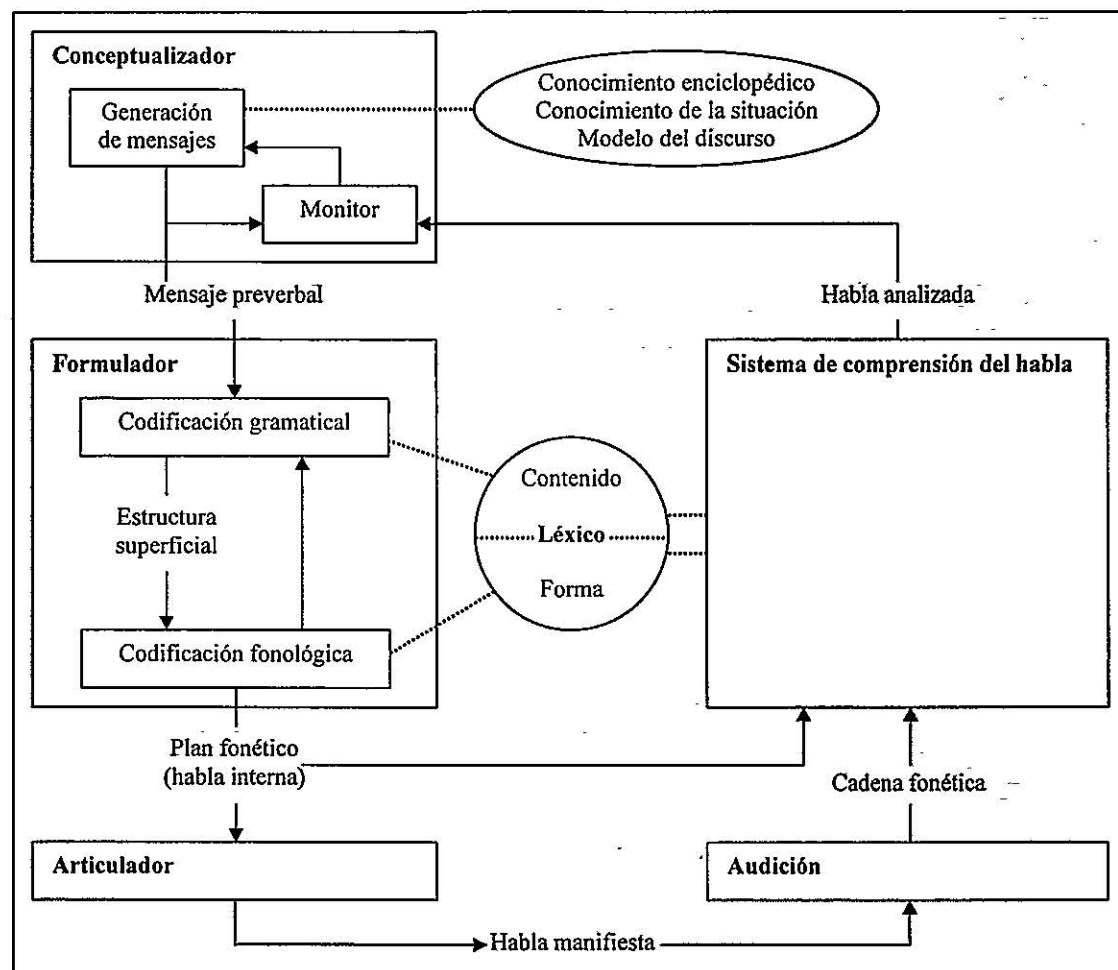


Figura 2.1.—Esquema del modelo de producción de Levelt (1989).

Por su parte, los procesos de microplanificación son los encargados de perfilar algunos aspectos estructurales de estos enunciados individuales. Por ejemplo, fijar el valor de la información nueva y dada, asignar el índice de accesibilidad de los referentes y determinar la perspectiva de la presentación son algunas de las tareas a realizar por los procesos de microplanificación. Con esto, se establece la forma lingüística de presentación de los referentes, el acto de habla, el foco del enunciado y su conjugación verbal. Este tipo de información estructural junto a la representación proposicional del contenido semántico de la oración configuran el mensaje preverbal que constituye la representación de salida del conceptualizador y de entrada del formador.

Queda por destacar, en cuanto al conceptualizador, que en él intervienen conocimientos generales del hablante tanto del mundo y de sí mismo como de la situación de interacción y el

modelo del discurso. Entre éstos están los conocimientos acerca del uso del lenguaje sin que por ello impliquen la participación de conocimientos específicamente lingüísticos.

Es en el segundo componente del sistema, el formulador, en el que intervienen los conocimientos lingüísticos que el hablante posee de una lengua particular. Esto es, los conocimientos acerca de la forma y el contenido de las oraciones que constituyen una lengua, en sus aspectos semánticos, sintácticos, léxicos, morfológicos y fonológicos. Es la aplicación de estos conocimientos lo que permite a los procesos de formulación lingüística realizar la función de traducir el mensaje preverbal al formato lingüístico. Este es un procedimiento de especificación progresiva de las distintas unidades estructurales que configuran una oración para, finalmente, obtener una secuencia de unidades lingüísticas mínimas que constituye, en los casos de lenguaje oral, un plan fonético.

Para ello, el formulador cuenta con la participación de dos grupos de procesos de codificación. Por un lado están los procesos encargados de la construcción de una estructura secuencial de unidades léxicas, conocida como la estructura superficial de la oración. Estos son los procesos de codificación gramatical que operan fundamentalmente con información sintáctica y léxica para crear el marco sintáctico y recuperar las unidades léxicas que se ajusten a las especificaciones conceptuales y pragmáticas contenidas en el mensaje preverbal.

Por otro lado, los procesos de codificación fonológica tienen la misión de traducir la estructura superficial de la oración a un plan fonético que constituye la representación de salida del formulador y de entrada del articulador. En ella están especificadas las secuencias de fonemas organizadas en unidades articulatorias y la información prosódica del enunciado lingüístico.

Finalmente, el plan fonético es traducido por el articulador a un plan motor que se compone de una serie de órdenes motoras para la emisión de los sonidos por el aparato fonador.

Así, la producción del lenguaje es descrita como un sistema de carácter jerárquico con tres fases de procesamiento principales. Entre las características más destacables podemos mencionar, por un lado, la separación de los mecanismos de procesamiento y las bases de conocimiento, y por otro lado, la incorporación del sistema de comprensión en la descripción de los procesos de producción. Asimismo, respecto a los procesos de formulación, podemos hacer

referencia a su carácter específicamente lingüístico y el papel central del léxico en esta fase del sistema.

No obstante, es importante resaltar que esta propuesta es un esquema general y no un modelo teórico particular. Como tal, no pretende establecer conclusiones definitivas acerca de la naturaleza de los procesos de producción. La utilidad de esta caracterización funcional reside, únicamente, en su capacidad para integrar las distintas cuestiones planteadas en el estudio de la producción verbal en un marco de referencial general. Es más, cada uno de los componentes y subcomponentes identificados requieren de especificaciones explícitas tanto de las representaciones y los procesos implicados como el momento y el modo de su intervención. Este es el objetivo último de la psicolingüística, construir modelos teóricos que contemplen estas cuestiones.

A estas consideraciones teóricas acerca de los procesos de codificación gramatical nos referiremos a continuación, en el siguiente epígrafe.

2.4. Codificación gramatical

Uno de los componentes básicos que configuran el sistema de producción verbal es el formulador. Su función consiste en dar al contenido comunicativo un formato lingüístico específico, esto es, traducir el mensaje a un código lingüístico particular. En esta fase de la producción intervienen los conocimientos que un usuario tiene de la lengua tanto en sus aspectos gramaticales como, en el caso del lenguaje hablado, sus aspectos fonológicos. Esta distinción se establece igualmente entre los procesos implicados en la formulación lingüística.

Los procesos de codificación gramatical son los encargados de construir la representación de la estructura superficial de una oración a partir de su representación conceptual. Consiste, básicamente, en la selección de unidades léxicas apropiadas y su ensamblaje en el marco sintáctico adecuado. Dichos procesos han sido objeto de múltiples estudios empíricos, cuyos resultados han contribuido a una mejor comprensión de los mismos (véase Igoa y García-Albea, 1999).

Es en estos aspectos de la producción verbal donde la psicolingüística cuenta con las propuestas teóricas más sólidas. Actualmente, el modelo de codificación gramatical más

representativo es el propuesto por Bock y Levelt (1994). De su exposición nos ocuparemos a continuación.

En este modelo se distinguen dos niveles de procesamiento en la codificación gramatical: el nivel funcional y el nivel posicional, distinción previamente introducida por Garrett (1975, 1980, 1988) en su modelo de producción de oraciones. En el procesamiento funcional se produce el acoplamiento de las representaciones conceptuales del mensaje con las representaciones gramaticales y léxicas más abstractas del enunciado. Para ello, intervienen dos grupos de procesos: los procesos de selección léxica y los procesos de asignación funcional. En la selección léxica son identificados los conceptos léxicos que expresan el contenido semántico del mensaje y es recuperada la información gramatical representada en el *lemma* asociado a cada concepto léxico. Asimismo, los procesos de asignación funcional proceden a la codificación sintáctica de la información temática con la asignación de las funciones que desempeñarán los participantes o papeles temáticos del mensaje en el marco sintáctico del enunciado.

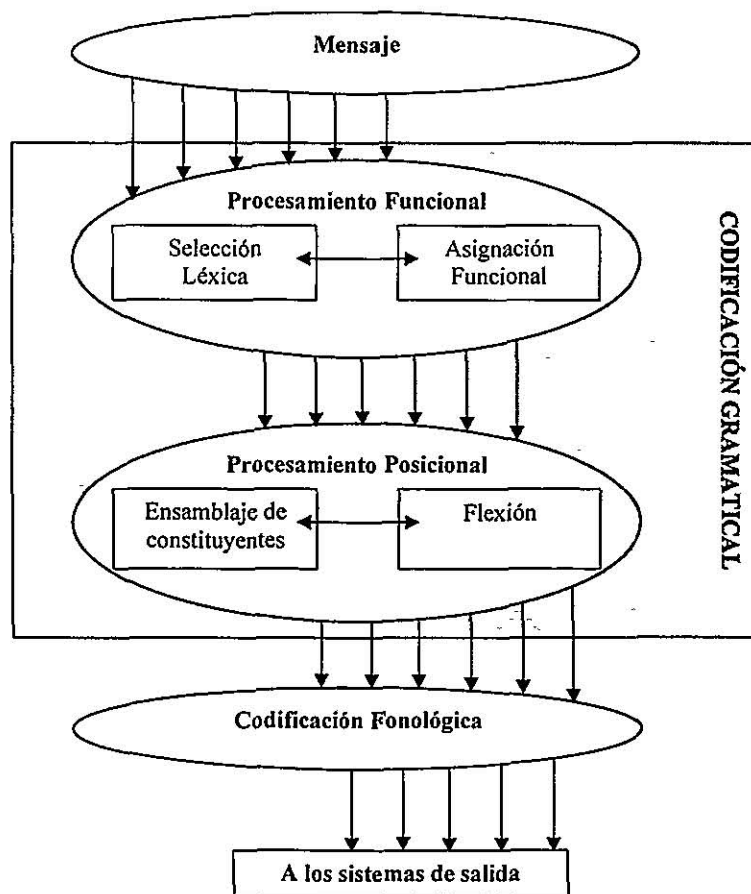


Figura 2.2.— Componentes del sistema de producción del lenguaje, adaptado de Bock y Levelt (1994) (Igoa y García-Albea, 1999).

Por su parte, en el procesamiento posicional se establece el orden de los elementos que componen la oración. Por un lado están los procesos de ensamblaje de constituyentes. Estos procesos son los encargados de especificar el orden de los componentes sintácticos y las dependencias existentes entre ellas, creando, así, una estructura jerárquica que controla el orden de producción de las palabras. Por otro lado, a partir de los *lemmas* seleccionados, son recuperados los *lexemas* en los que están representados las formas morfológicas y fonológicas de los elementos léxicos. Los procesos de flexión proceden, entonces, a la asignación de sufijos y partículas gramaticales que marcan las propiedades morfológicas de las palabras.

Vamos a ilustrar, con un ejemplo, los distintos procesos postulados para la codificación gramatical. En la formulación del enunciado "ella le pasó la sal", el sistema de codificación gramatical recibe como entrada una representación conceptual del mensaje que, presumiblemente, contiene información acerca de una acción-evento pasada en el que un actor-agente femenino transfiere un objeto-tema específico a un recipiente-benefactivo humano masculino.

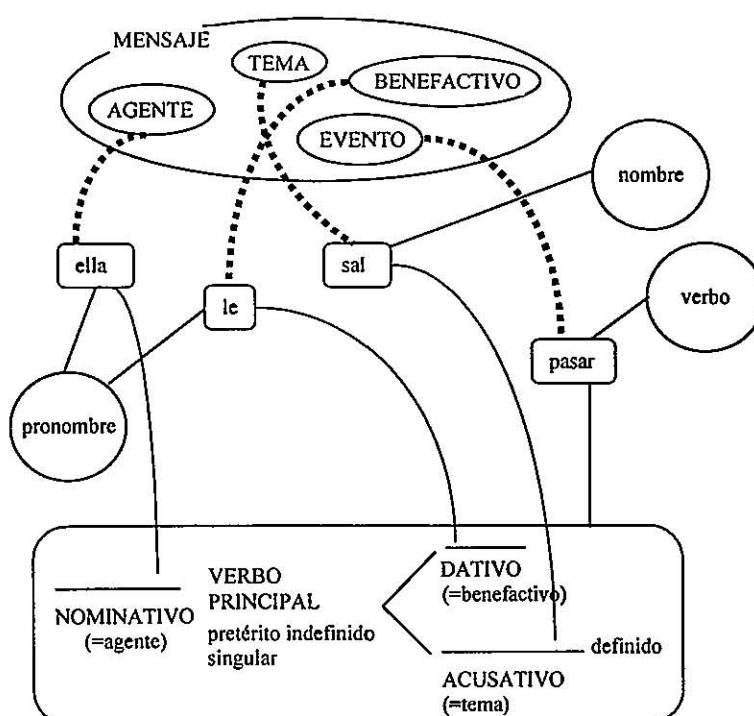


Figura 2.3.—Productos del procesamiento funcional (adaptado de Bock y Levelt, 1994).

Los procesos de selección léxica recuperan los *lemmas* de los pronombres femenino (*ella*) y masculino (*le*), un nombre (*sal*) y un verbo (*pasar*) para cada uno de los papeles temáticos

representados en el mensaje, agente, benefactivo, tema y evento, respectivamente. Cada uno de los *lemmas* recuperados son vinculados con una función sintáctica por los procesos de asignación funcional. Así, el *lemma* del pronombre femenino es vinculado con la función nominativa (sujeto), el masculino con la función dativa (objeto indirecto), la *sal* con la función acusativa (objeto directo) y *pasar* con la función del verbo principal. En la Figura 2.3. tenemos una representación gráfica del procesamiento funcional aplicada a nuestro ejemplo.

Ya en el nivel de procesamiento posicional, las funciones sintácticas son organizadas en una estructura jerárquica por los procesos de ensamblaje de constituyentes según sus relaciones de dependencia. Algunos de los nodos de dicha estructura, requiere una elaboración adicional. Estos son los casos en que los procesos de flexión proceden a completar la estructura secuencial de los marcos sintácticos con marcos morfológicos y léxicos. En el caso del marco sintáctico vinculado con la función del verbo principal, la expresión de modo y tiempo verbal en el verbo "pasó" requiere la intervención de los procesos de flexión para crear una nueva estructura que vincula las unidades morfológicas raíz y sufijo con la función sintáctica correspondiente. Otro ejemplo lo encontramos en la especificación adicional que acompaña a la función acusativa para su adecuada realización. Esto es, la expresión del carácter definido del nombre "sal" requiere de un elemento léxico adicional. Así, del marco sintáctico vinculado con la función acusativa se deriva de dos nodos adicionales que corresponden a los marcos léxicos del artículo definido "la" y del nombre "sal".

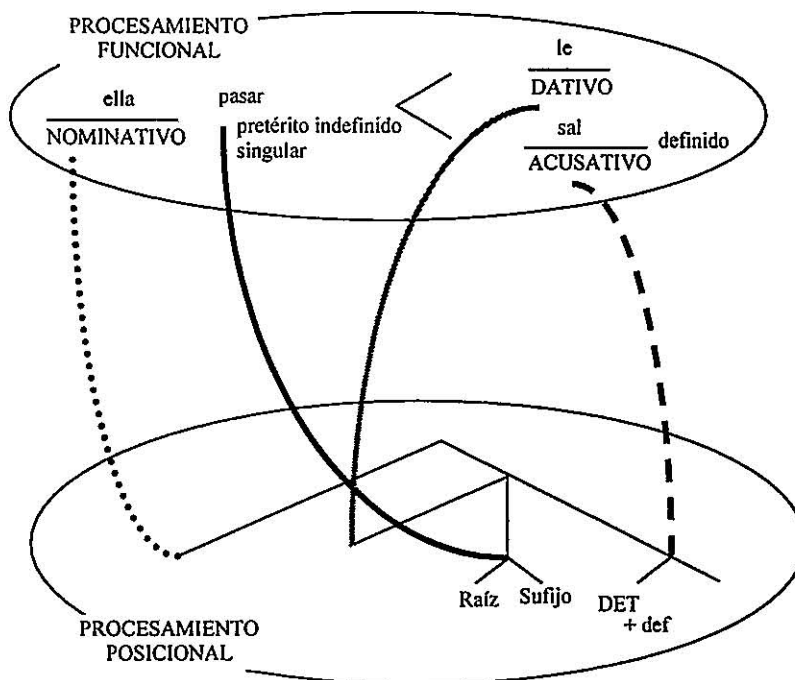


Figura 2.4.— Esquema ilustrativo de la codificación gramatical (adaptado de Bock y Levelt, 1994).

Con todo, los procesos de la codificación gramatical generan una representación de la estructura superficial de la oración. Sobre esta representación intervienen los procesos de codificación fonológica que le dan el formato apropiado para su emisión oral.

Hasta el momento, hemos descrito las funciones a realizar por los procesos implicados en la codificación gramatical. Sin embargo, como modelo explicativo, exige una definición explícita acerca de cómo el sistema cognitivo lleva a cabo dichas tareas. Para ello, estos autores completan su propuesta con diversas consideraciones empíricas y teóricas, integrando modelos computacionales explicativos de algunos de sus procesos, como por ejemplo, el modelo de selección léxica de Roelofs (1992) y los modelos de Yngve (1960) y de Smedt (1990) como posibles explicaciones de los procesos de ensamblaje de constituyentes.

Para concluir nuestra exposición del modelo, quedan por destacar los supuestos adoptados acerca de la naturaleza de estos procesos. Por un lado, el carácter modular de los niveles de procesamiento. Esto es, que cada uno de ellos está influenciado únicamente por el nivel de procesamiento inmediatamente superior. Así, los procesos de selección léxica y de asignación funcional están controlados por la información representada en el nivel de mensaje pero no están afectados por las propiedades fonológicas de las palabras.

Por otro lado, este modelo asume que la producción del lenguaje es un proceso incremental, es decir, que los procesos de un nivel de procesamiento no necesitan que el nivel inmediatamente superior concluya sus tareas para iniciar su actividad. En esto, el modelo requiere de una especificación precisa del cómo y cuándo han de intervenir cada uno de los procesos en la codificación gramatical del mensaje.

Es en los modelos computacionales donde estas cuestiones adquieren especial relevancia. Asimismo, en este tipo de aproximación teórica, la definición de las estructuras representacionales participantes constituye una de las tareas principales a realizar. Por tanto, para ofrecer un panorama más completo de lo expuesto hasta el momento, creemos conveniente realizar una aproximación más detallada a uno de los modelos computacionales más significativos dentro de la psicología del lenguaje, nos referimos al modelo de formulación lingüística de Schlesinger (1977).

Dos elementos básicos describen este modelo: los *marcadores de entrada* y las *reglas de realización*. Así, la transformación de las representaciones conceptuales en oraciones específicas se lleva a cabo aplicando reglas de realización sobre los marcadores de entrada.

Los marcadores de entrada son constructos teóricos definidos para la formalización de las representaciones conceptuales. Son expresiones compuestas por *elementos protoverbales* (unidades precursoras de palabras) y relaciones que entre ellos se establecen. Consideremos la siguiente expresión:

(AGENT-ACTION *Juan*, (INSTRUMENT-ACTION (POSSESSOR-POSSESSED *David*, (PLURALITY *apunte*)), *estudiar*)).

Este es un ejemplo de un marcador de entrada que contiene cuatro elementos protoverbales (*Juan*, *David*, *apunte*, *estudiar*) y cuatro tipo de relaciones (PLURALITY, AGENT-ACTION, INSTRUMENT-ACTION, POSSESSOR-POSSESSED). Por ejemplo, la etiqueta POSSESSOR-POSSESSED denota una relación de posesión con dos argumentos. El primero de ellos, el poseedor, tiene por valor un elemento protoverbal (*David*) mientras que el segundo es, en sí mismo, una relación (PLURALITY) que a diferencia de las demás, tiene un elemento protoverbal como su único argumento (*apunte*).

Para convertir este tipo de expresiones en enunciados lingüísticos específicos, Schlesinger propone un conjunto de reglas de realización de distintas clases. En primer lugar, nos referiremos a las reglas de relación. Son expresiones ($a \rightarrow b$) donde el componente izquierdo representa las relaciones que aparecen en los marcadores de entrada y el componente derecho, las operaciones de realización para dichas relaciones. Para nuestro ejemplo, las reglas de relación a aplicar son:

- (1) AGENT-ACTION $a, b \rightarrow (\text{AGENT } a^N + b^V)^S$
- (2) INSTRUMENT-ACTION $a, b \rightarrow (\text{INST } b^V + \text{"con"} + a^N)^V$
- (3) POSSESSOR-POSSESSED $a, b \rightarrow (\text{POSS } b^N + \text{"de"} + a^N)^N$
- (4) PLURALITY $a \rightarrow (\text{PLUR } a^N + \text{"s"})^N$

Como podemos observar, cada elemento protoverbal y su relación reciben una categoría gramatical (los superíndices). Asimismo, se establece la posición relativa de los elementos (+) y se introducen los afijos flexivos ("s") y las expresiones gramaticales ("con", "de").

Siguiendo con el ejemplo, una posible secuencia de aplicación de estas reglas sobre el marcador de entrada es el siguiente:

Paso 1. Relación del marcador de entrada realizada: INSTRUMENT-ACTION

Resultado de la aplicación de la regla (2):

(AGENT-ACTION *Juan*, (INST *estudiar*^V + "con" + (POSSESSOR-POSSESSED *David*, (PLURALITY *apunte*))^N)^V)

Paso 2. Relación del marcador de entrada realizada: POSSESSOR-POSSESSED

Resultado de la aplicación de la regla (3):

(AGENT-ACTION *Juan*, (INST *estudiar*^V + "con" + (POSS (PLURALITY *apunte*)^N + "de" + *David*^N)^N)^V)

Paso 3. Relación del marcador de entrada realizada: AGENT-ACTION

Resultado de la aplicación de la regla (1):

(AGENT *Juan*^N, (INST *estudiar*^V + "con" + (POSS (PLURALITY *apunte*)^N + "de" + *David*^N)^N)^V)^S

Paso 4. Relación del marcador de entrada realizada: PLURALITY

Resultado de la aplicación de la regla (4):

(AGENT *Juan*^N, (INST *estudiar*^V + "con" + (POSS (*apunte*^N + "s")^N + "de" + *David*^N)^N)^V)^S

Con la aplicación de las reglas de relación, a todos los elementos les son asignadas una posición y una categoría gramatical. De igual modo, todas las palabras de función y los afijos flexivos necesarios para la expresión de las relaciones son incorporados por estas reglas de realización a su estructura representacional.

Por otro lado, todos los elementos protoverbiales deben ser reemplazados por unidades léxicas apropiadas. Esta función la realizan las reglas de lexicalización. Un requisito adicional a tener en cuenta es que las palabras seleccionadas deben pertenecer a la categoría gramatical asignada a los elementos protoverbiales. Sin embargo, en los casos en que este requisito no se cumpla, la lexicalización demandará una revisión del proceso.

Finalizada la realización de las reglas relacionales y la lexicalización, operan las reglas de concordancia para establecer, como su propio nombre indica, las concordancias gramaticales. Con esto concluye la codificación gramatical de la que se obtiene una secuencia ordenada de ítems léxicos sobre los cuales operan las reglas fonológicas y de entonación para generar su emisión oral.

2.5. Recapitulación

La codificación gramatical es el componente del proceso de producción verbal encargado de transformar una representación conceptual del mensaje (*i.e.*, mensaje preverbal) a un formato lingüístico específico. Para ello, el sistema cognitivo lleva a cabo un conjunto de operaciones. El modelo propuesto por Bock y Levelt (1994) distingue cuatro grupos de operaciones que intervienen en este proceso de codificación: asignación funcional, selección léxica, ensamblaje de constituyentes y flexión. Cada uno de ellos son descritos en términos de las funciones que

realizan en el proceso y, de forma individual y en distinta medida, definidos por las operaciones que intervienen en su realización.

Sin embargo, no es sino en el modelo computacional de Schlesinger (1977) donde podemos encontrar una especificación precisa del funcionamiento del proceso de codificación gramatical. El modelo cuenta con una definición formal de la estructura de representación inicial y describe el proceso de codificación como una secuencia de aplicación sucesiva de reglas de transformación de la representación de entrada. En esto último observamos la principal deficiencia del modelo. El proceso de transformación al operar sobre una única estructura de representación funciona como un componente de todo o nada. Esto es, la representación de salida que sirva de entrada para el siguiente proceso sólo se obtiene finalizado todo el proceso de transformación. Este aspecto del modelo es contrario a la propuesta de Bock y Levelt (1994) acerca de la naturaleza incremental del proceso basada en numerosos datos empíricos.

Con objeto de llegar a modelos formales que subsanen las deficiencias observadas en las propuestas previas, la psicolingüística debe considerar los recursos que desde la inteligencia artificial se ofrecen para el análisis de nuestro objeto de estudio: la codificación gramatical. De ello nos ocuparemos en el próximo capítulo.

Capítulo 3: Procesamiento de lenguaje natural

3.1. Introducción

Las posibilidades que los ordenadores ofrecen al estudio de las funciones lingüísticas fueron vislumbradas desde su invención. Su capacidad para la manipulación de símbolos se propone especialmente adecuada para abordar estas cuestiones. Así, nace la idea de la traducción mecánica.

Esta idea de traducir textos de una lengua a otra mediante el uso de los ordenadores fue concebida a principio de la década de los años 30 por P. P. Smirnov-Troyansky y G.B. Artsouni (véase Barr y Feigenbaum, 1981). Sin embargo, no es hasta finales de los 40, con la aparición de los ordenadores digitales, cuando se produce la génesis real de la traducción mecánica con los trabajos de Warren Weaver (1955) y A. Donald Booth (1948). Según el planteamiento de estos autores, la tarea de traducción consiste, básicamente, en consultar un diccionario bilingüe, reemplazar cada palabra por su equivalente en la lengua destino y, de la cadena de palabras resultante, organizarlas en el orden apropiado. Así, desde el convencimiento de que el problema principal de la traducción mecánica está en la incorporación al ordenador de un diccionario completo de al menos dos lenguas, centraron sus esfuerzos, fundamentalmente, en la mecanización del vocabulario. Por supuesto, eran conscientes que con contar simplemente con un diccionario no se resolvían todos los problemas.

Los sucesivos fracasos de los programas de traducción palabra por palabra, demostraron la complejidad de la tarea que va más allá de un diccionario de consulta y del reordenamiento de las palabras. Como por ejemplo, las ambigüedades y los significados múltiples, habitualmente presentes en el uso del lenguaje natural, requieren, para su resolución, de un modelo del mundo y de la capacidad de manipular los símbolos de ese modelo. Es más, no nos es ajeno el hecho de que personas totalmente dotadas de este conocimiento son incapaces de traducir a su propia lengua desde otra desconocida utilizando únicamente un diccionario.

Estos aspectos de deficiencia en el planteamiento de la traducción mecánica estaban ya contemplados en la memoria de investigación de Weaver, "Translation", en 1949. En ella

esboza la idea de que traducir de una lengua A a una lengua B supone la participación de un lenguaje intermedio universal, el "interlingua", compartido por todas las lenguas. Esto es, una representación semántica del mensaje. Por tanto, la traducción mecánica exige del programa la capacidad de comprensión y, para ello, es imprescindible el conocimiento del mundo.

A partir de los años 60, la tarea de traducción mecánica ha sido abordada desde una nueva perspectiva. Esto es, la concepción del lenguaje humano como una capacidad cognitiva compleja en la que participan diferentes clases de conocimiento, lingüísticos y no lingüísticos. Esta aproximación de la inteligencia artificial al estudio del lenguaje es conocida como el procesamiento de lenguaje natural.

3.2. Procesamiento de lenguaje natural

El procesamiento de lenguaje natural es el área de investigación de la inteligencia artificial dedicada al estudio de sistemas computacionales que analizan, comprenden, procesan y producen lenguaje natural. Tiene el doble objetivo de desarrollar sistemas computacionales prácticos (traducción mecánica, interfaz con sistemas de base de datos, sistema de resumen de textos, etc.) y de contribuir a la comprensión del procesamiento lingüístico humano (desarrollando modelos computacionales y creando programas para su instanciación).

Se desarrollan de forma relativamente independiente el estudio de la comprensión y de la generación. En comprensión, el objetivo es analizar textos en lenguaje natural, y extraer su significado de forma que, representado en un cierto formato, pueda ser procesado por el ordenador para hacer uso de ello.

La generación de lenguaje natural consiste en la producción de textos en lenguaje natural realizado por un sistema computacional a partir de una intención comunicativa, esto es, de una meta a alcanzar por medio de un acto comunicativo. Es considerado un proceso de toma de decisiones que, a partir de unas especificaciones de entrada, produce de forma automática enunciados en lenguaje natural que pueden ser desde oraciones individuales a discursos completos. Así, un proceso completo de generación de discurso debe tener en cuenta las siguientes cuestiones (Dale, Mellish y Zock, 1990):

- ♦ ¿cuál es el contenido del discurso?
- ♦ ¿cómo se organiza ese contenido en un discurso coherente?
- ♦ ¿cómo se divide ese contenido en oraciones y cláusulas?
- ♦ ¿qué construcciones sintácticas se aplican?
- ♦ ¿cómo se describen las entidades conceptuales?
- ♦ ¿qué palabras se utilizan para referirse a ellas?

Como podemos observar, es un problema complejo, por lo que se han establecido subparcelas más manejables para facilitar su estudio, distinguiendo dos niveles de procesamiento: estratégico y táctico (Thompson, 1977).

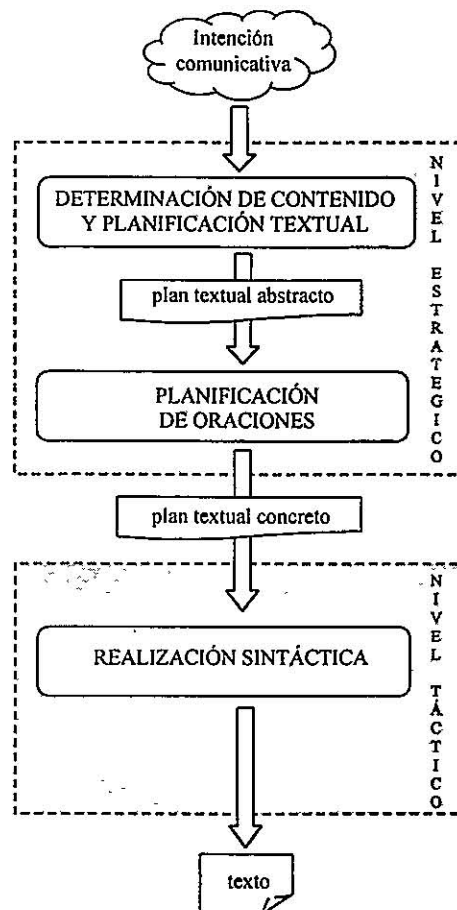


Figura 3.1.—Arquitectura general de generadores de lenguaje natural (Thompson, 1977 y Reiter, 1994).

La generación estratégica hace referencia al qué decir. En este nivel de procesamiento se utilizan fundamentalmente información no lingüística (conocimiento del mundo, contexto previo del discurso, intenciones del oyente, etc.) para, por un lado, identificar las metas que se

pretenden alcanzar con el discurso y planificar la forma de lograrlo y, por otro lado, determinar el contenido proposicional y su organización en un plan textual (*e.g.*, esquemas de McKeown, 1982, 1985) así como su descomposición en unidades lingüísticas realizables (*e.g.*, relaciones retóricas de Mann y Thompson, 1987a, 1987b).

Las cuestiones relativas al cómo decir son consideradas en la generación táctica. Consiste, básicamente, en la selección léxica y gramatical para llevar a cabo el plan textual recibido. En este nivel de procesamiento es imprescindible la participación del conocimiento de los recursos del lenguaje natural en el que se va a construir la forma lingüística superficial, esto es, su gramática y su léxico.

Desde una perspectiva técnica, Reiter (1994) estableció una arquitectura de tres niveles. En un primer nivel se decide la información a transmitir en el texto y se establece la organización de dicha información en una estructura retórica coherente. Estos son los procesos de determinación de contenido y planificación textual. En un segundo nivel se decide la organización de la información en párrafos y oraciones individuales, introduciendo los recursos de agregación y cohesión adecuados para su realización superficial. Estos dos primeros niveles corresponden a la generación estratégica.

La realización sintáctica, equivalente al procesamiento táctico, procede a la generación de oraciones individuales gramaticalmente correctas a partir de la representación de su contenido semántico. En esto, la gramática de lenguaje natural constituye su recurso principal. Por lo que investigadores de este área de procesamiento de lenguaje natural han desarrollado distintas técnicas de codificación y utilización de la información gramatical en sistemas de generación táctica.

3.3. Técnicas de codificación y utilización de información gramatical

La gramática de un lenguaje se define como el esquema en el que se especifican las oraciones permitidas en dicho lenguaje. Asimismo, contiene las reglas sintácticas para la formación correcta de las distintas estructuras sintagmáticas. Existen diversos recursos básicos para su descripción formal (véase Barr y Feigenbaum, 1981; Grosz, Sparck Jones y Webber, 1986; Kempen, 1987; Beardon, Lumsden y Holmes, 1991). En este apartado presentaremos los

principales formalismos gramaticales y computacionales utilizados para la construcción de generadores de lenguaje natural.

3.3.1. Gramáticas libres de contexto

La teoría de lenguajes formales introducida por Noam Chomsky (1956, 1959b) es una de las aportaciones más importantes realizadas al estudio del lenguaje. En ella Chomsky definió cuatro tipos de gramáticas formales como modelos potenciales para la descripción de los lenguajes naturales. Estas gramáticas describen la estructura de un lenguaje por medio de un conjunto de símbolos terminales (el léxico), de un conjunto de símbolos no terminales (las categorías sintácticas) y de un conjunto de reglas de producción que especifican las relaciones entre secuencias de símbolos terminales y no terminales. De entre estos últimos uno se distingue por ser el símbolo de partida.

Así, una gramática G es definida por una tupla $G = (N, \Sigma, P, S)$, donde

- ♦ N es el conjunto de símbolos no terminales,
- ♦ S es el elemento del conjunto N distinguido como símbolo de partida,
- ♦ Σ es el conjunto de símbolos terminales que junto con N conforman el vocabulario $V = N \cup \Sigma$ de la gramática, y
- ♦ P es el conjunto de producciones de la forma $X \rightarrow Y$, donde X e Y son secuencias finitas de elementos del conjunto V (cadena), y X no debe ser una cadena vacía.

Sobre la forma de las producciones se establecen restricciones sucesivas que dan lugar a los cuatro tipos de gramáticas formales. La gramática *Tipo 0* es la más general de las clases de gramática y no recibe ninguna restricción sobre la forma de las producciones.

Una primera restricción aplicada a la forma de las producciones es aquella que establece que en toda relación $X \rightarrow Y$, entre el conjunto de elementos de Y ha de aparecer al menos uno de los elementos de X . Una definición alternativa pero equivalente a este tipo de gramáticas es que las producciones deben tener la forma $uXv \rightarrow uYv$, donde X es un símbolo no terminal, u y v son cadenas arbitrarias, posiblemente vacías, de elementos de V , e Y es una cadena no vacía de elementos de V . Por tanto, podemos decir que en las gramáticas *Tipo 1* o *gramáticas sensibles*

al contexto, X produce Y sólo en el contexto de u y v . Como ejemplo vamos a ver una gramática que genera el lenguaje L , siendo $L(G) = \{a^n b^n c^n / n \geq 0\}$:

$$\begin{aligned} S &\rightarrow aSBC \\ S &\rightarrow aBC \\ CB &\rightarrow BC \\ aB &\rightarrow ab \\ bC &\rightarrow bb \\ bC &\rightarrow bc \\ cC &\rightarrow cc \end{aligned}$$

Aplicando esta gramática obtenemos el conjunto de cadenas abc , $aabbcc$, $aaabbbccc$, etc. Veamos:

$$\begin{array}{ccccc} (S \rightarrow aBC) & (aB \rightarrow ab) & (bC \rightarrow bc) & & \\ \downarrow & \downarrow & \downarrow & & \\ S & \Rightarrow & aBC & \Rightarrow & abC & \Rightarrow & abc \end{array}$$

Las gramáticas libre de contexto o Tipo 2 son aquellas en las que en cada producción $X \rightarrow Y$, X es un único símbolo del conjunto N , el conjunto de los símbolos no terminales. Con esto, las producciones pueden aplicarse sobre un símbolo independientemente de la cadena en la que está incluido. Un ejemplo de una gramática libre de contexto que puede ser utilizada para generar algunas oraciones en lenguaje natural puede ser definida así:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle NP \rangle \langle VP \rangle \\ \langle NP \rangle &\rightarrow \langle DET \rangle \langle N \rangle \\ \langle NP \rangle &\rightarrow \langle N \rangle \\ \langle VP \rangle &\rightarrow \langle V \rangle \langle NP \rangle \\ \langle DET \rangle &\rightarrow \text{los} \\ \langle N \rangle &\rightarrow \text{alumnos} \\ \langle N \rangle &\rightarrow \text{tienen} \\ \langle V \rangle &\rightarrow \text{exámenes} \end{aligned}$$

El uso de este tipo de gramáticas en programas de lenguaje natural cuenta con una importante propiedad de las gramáticas libres de contexto, y es que todas las derivaciones pueden ser convenientemente representadas como un árbol en el que se visualiza la estructura sintagmática de las oraciones. Así, aplicando las reglas definidas en el ejemplo, la oración “*los alumnos tienen exámenes*” estaría representada por el siguiente árbol de derivación:

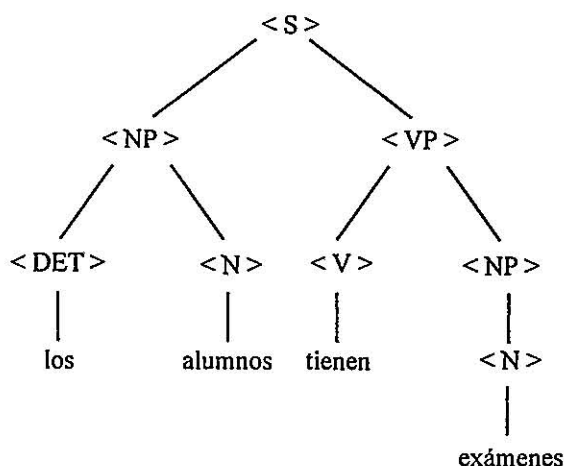


Figura 3.2.—Árbol sintagmático de derivación de la oración "los alumnos tienen exámenes".

Esta propiedad está presente, también, en las gramáticas sensibles al contexto. Y es por esto que las gramáticas libres de contexto y las gramáticas sensibles al contexto son conocidas, a su vez, como gramáticas sintagmáticas.

Finalmente, si todas las producciones que definen la gramática tienen la forma $X \rightarrow aY$ o $X \rightarrow a$, donde X e Y son un único símbolo del conjunto V y a un símbolo terminal, entonces es una gramática Tipo 3 o una gramática regular.

En esto podemos observar la difícil tarea con que los investigadores del procesamiento de lenguaje natural se enfrentan a la hora de desarrollar formalismos adecuados para describir y procesar lenguajes naturales. Tienen que conjugar el hecho de que las gramáticas formales generales (Tipo 0 y Tipo 1) son difíciles de manejar computacionalmente y el hecho de que el lenguaje natural se describe mejor, aunque no de forma exclusiva, por medio de gramáticas más potentes que las gramáticas libres de contexto.

Actualmente, los formalismos que se utilizan exceden con mucho la capacidad generativa de las gramáticas libres de contexto. A estos formalismos se denominan gramáticas libres de contexto *aumentadas*. Su capacidad computacional radica en el hecho de que emplean los mecanismos inherentes a las gramáticas libres de contexto para expresar la información sintáctica de un lenguaje y utilizan algoritmos y técnicas directamente relacionados con ellas para su procesamiento. Sobre esta base se introducen propiedades formales y computacionales

adicionales que contribuyen a aumentar su poder generativo. Las redes de transición aumentada y las gramáticas de unificación son ejemplos de lo comentado.

3.3.2. Gramática transformacional

Otra de las aportaciones de Chomsky, la teoría de la gramática generativa (Chomsky, 1957), ha influido radicalmente en el estudio del lenguaje en varias disciplinas: en la lingüística, en la psicología y en la inteligencia artificial. En ella, Chomsky propone reglas de transformación para solventar las deficiencias observadas en las gramáticas libres de contexto para la descripción estructural de lenguajes naturales.

En su versión estándar, el procedimiento de generación de oraciones comienza con la elaboración de una representación de la estructura profunda de las oraciones mediante la aplicación de reglas de gramática libre de contexto. Sobre estas estructuras operan las reglas de transformación para dar lugar a las representaciones de su estructura superficial.

Sin embargo, las gramáticas transformacionales no se muestran eficientes para la construcción de sistemas computacionales de generación de lenguaje natural. Este hecho es considerado ya por el propio autor cuando plantea esta gramática como método descriptivo y no funcional. Asimismo, los diversos intentos de verificar la realidad psicológica de las transformaciones resultaron bastante infructuosos (Fodor, Bever y Garrett, 1974; Bresnan, 1978, 1981; Gazdar, 1981). Por esto, el uso de la gramática transformacional como base para el desarrollo de un generador, sólo puede ser aceptado en los casos en que no se consideran como cuestiones de importancia la validez psicológica del sistema y su eficiencia computacional.

3.3.3. Redes de transición aumentada

Una de las primeras formalizaciones sistemáticas de gramáticas libres de contexto aumentada son las redes de transición aumentada, desarrolladas inicialmente por William Wood (1970) al incorporar propiedades adicionales a las redes de transición recursiva.

Las redes de transición recursivas son un recurso teórico que permite la representación de los distintos procedimientos a seguir en la realización de una tarea determinada. Están definidas por un conjunto finito de nodos o estados vinculados entre sí mediante arcos o líneas con

flechas direccionales. De entre los nodos uno es designado como inicial y al menos otro como final. Veamos un sencillo ejemplo:

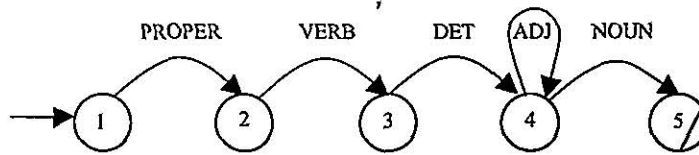


Figura 3.3.—Ejemplo 1 de red de transición recursiva.

En este caso, los arcos son denominados por categorías de palabras del lenguaje natural. Supongamos los siguientes valores a estas categorías:

PROPER: *Juan, María*
 VERB: *tiene, sabe*
 DET: *la, una*
 ADJ: *difícil, hermosa*
 NOUN: *lección, casa*

Con esto, esta gramática especifica oraciones como:

Juan tiene una casa
Juan tiene una hermosa lección
Juan tiene una hermosa difícil casa
María sabe la lección
María sabe la hermosa hermosa casa
etc.

Ahora bien, las transiciones entre nodos no sólo pueden ser denominadas por identificadores de elementos terminales, como son las categorías de palabras, sino que también pueden ser denominadas por identificadores de redes de transición. Por ejemplo:

S:

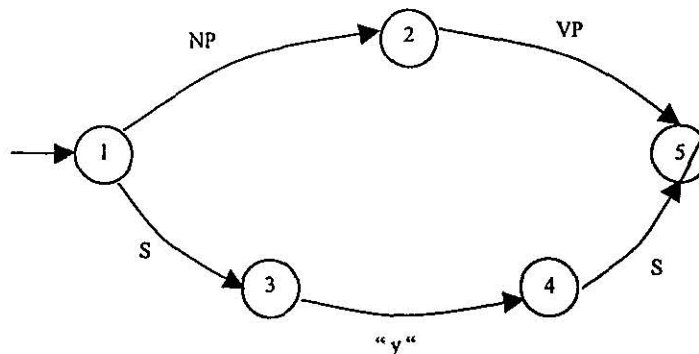
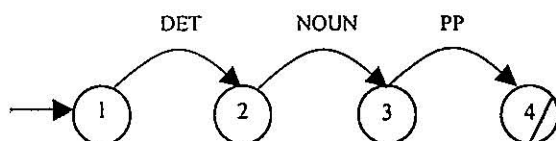


Figura 3.4.—Ejemplo 2 de red de transición recursiva.

En este ejemplo, están representados dos casos de la propiedad de recursividad. En la red etiquetada con *S* tenemos un primer ejemplo, en la que dos de sus arcos invocan a su propia definición. Las redes *NP* y *PP* representan un ejemplo de recursividad entre dos redes que interactúan dentro de un conjunto de redes de transición.

NP:



PP:

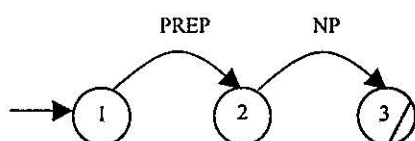


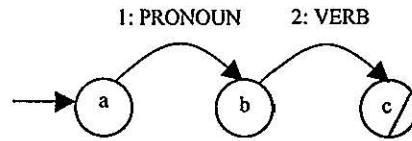
Figura 3.5.— Ejemplos 3 y 4 de red de transición recursiva.

Las redes de transición recursiva son una representación alternativa pero equivalente a las gramáticas libres de contexto, por lo que comparten las limitaciones de este tipo de gramáticas para la descripción de lenguajes naturales. Para solventar estas cuestiones, las redes de transición recursiva han sido mejoradas con parámetros y condiciones que guían la elección de recorridos. Estas son las redes de transición aumentada.

Una red de transición aumentada es una red de transición recursiva con tres propiedades adicionales:

- ♦ Cuenta con un conjunto de registros para almacenar información de los distintos estados de la red.
- ♦ Los arcos de la red pueden llevar asociadas condiciones que deben ser satisfechas para que una transición sea considerada viable.
- ♦ Ciertas acciones unidas a un arco, serán ejecutadas en la transición de los estados vinculados por dicho arco.

S:



Registro: Number = { singular, plural }

S1: Acción: Asignar Number con el valor del number de Word

S2: Condición: Comprobar Number igual al valor del number de Word

Lexicon:

él PRONOUN (number = singular)

ellos PRONOUN (number = plural)

tiene VERB (number = singular)

tienen VERB (number = plural)

Figura 3.6.—Ejemplo de red de transición aumentada.

En este ejemplo, **Word** es la variable que tiene como su valor la palabra a generar. En la transición del primer arco (S1) una palabra debe ser seleccionada de la categoría **PRONOUN**. El valor del atributo **number** del pronombre recuperado es asignado a la variable **Number** del registro. En la transición S2, una palabra de la categoría **VERB** debe ser seleccionada satisfaciendo la condición establecida de que su valor **number** debe ser el mismo que la variable **Number** del registro.

Las redes de transición aumentada fueron aplicadas con cierto éxito en sistemas como LUNAR (Woods, 1973) o BABEL (Goldman, 1975). Muchos otros sistemas, fundamentalmente de comprensión del lenguaje, han utilizado esta técnica para la representación de su conocimiento gramatical.

3.3.4. Unificación

Las redes de transición aumentada tienen un importante componente procedimental. Este carácter procedimental limita su flexibilidad y eficacia en algunos casos. Existen otras gramáticas sintagmáticas aumentadas que son fundamentalmente declarativas. Estas son las gramáticas de unificación (véase Allen, 1989).

La unificación es un concepto de la lógica predicativa que hace referencia al procedimiento de unión de dos expresiones predicativas idénticas. El procedimiento de unificación consiste, fundamentalmente, en la comparación de dos expresiones y en la búsqueda de sustituciones que las hagan idénticas. Por ejemplo, dadas las expresiones $P(x,x)$ y $P(y,z)$, el algoritmo de unificación procede de la siguiente manera:

- (a) Compara los predicados de la expresión; en este caso, son dos ocurrencias de P , por lo que podrían ser emparejados.
- (b) Compara los argumentos de dos en dos:
 - (b.1) Compara x e y y aplica la sustitución (y/x) a toda la expresión.
 - (b.2) Prosigue con el segundo argumento. Compara y y z y aplica la sustitución (z/y) .
- (c) Finaliza el proceso de comparación con la identificación de la composición de las sustituciones $(z/y)(y/x)$.

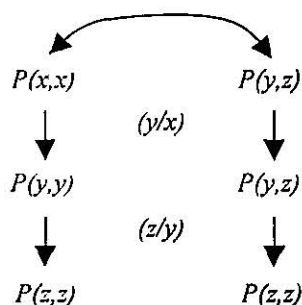


Figura 3.7.—Ejemplo de procedimiento de unificación.

En las gramáticas de unificación, la unificación se define como la operación que compone dos objetos lingüísticos para dar lugar a un único objeto. Los objetos lingüísticos se describen en términos de estructuras de rasgos, esto es, conjuntos de pares atributo-valor donde valor puede ser tanto un símbolo atómico como una estructura de rasgos. Así, si dos objetos lingüísticos son compatibles entonces el objeto resultante de la unificación contendrá los rasgos que ambos aportan para su descripción. En caso de que se decidan incompatibles, el resultado será una descripción vacía. Por ejemplo, dadas estas dos estructuras de rasgos:

[cat = np,
det = la,
head = [cat = noun, number = singular, gender = female]]

[cat = noun,
word = casa,
number = singular,
gender = female]

se podrían unificar creando la siguiente descripción:


```
[cat = np,
 det = la,
 head = [ cat = noun,
          word = casa,
          number = singular,
          gender = female]]
```

Una gramática de unificación prototípica es definida por reglas de gramáticas libres de contexto enriquecidas por un conjunto de especificaciones de rasgos de los símbolos gramaticales que aparecen tanto en las reglas como en el léxico asociado. Ejemplos de esta clase son la gramática léxico-funcional (Bresnan y Kaplan, 1982) y la gramática de estructura sintagmática generalizada (Gazdar, Klein, Pullum y Sag, 1985).

Como alternativa, se han desarrollado gramáticas de unificación que codifican la información gramatical, representada hasta el momento en las reglas gramaticales libres de contexto mediante restricciones funcionales especificadas por los valores reentrantes de las estructuras de rasgos. De esta forma, se ha homogeneizado el formalismo de representación de la información lingüística, en el que las relaciones entre objetos quedan incluidas dentro de la propia descripción de los objetos. A esta clase pertenecen dos de las gramáticas con mayor repercusión en la actualidad: la gramática de unificación funcional (Kay, 1979, 1985) y la gramática de estructura sintagmática núcleo-dirigido (Pollard y Sag, 1987, 1994).

A la gramática de unificación funcional haremos referencia posteriormente en la exposición del componente de generación táctica del sistema COMET (McKeown, Elhadad, Fukumoto, Lim, Lombardi, Robin y Smadja, 1990).

3.3.5. Gramáticas léxico-sintagmáticas

Usuario:	Todos los hombre son iguales.
ELIZA:	¿En qué sentido?
Usuario:	Siempre nos están fastidiando con una cosa u otra.
ELIZA:	¿Me puede dar un ejemplo específico?
Usuario:	Bueno, mi novio me hizo venir aquí.
ELIZA:	Su novio la hizo venir aquí.
Usuario:	Dice que casi siempre estoy deprimida.
ELIZA:	Siento que esté deprimida.
Usuario:	Es cierto. Soy infeliz.
ELIZA:	¿Piensa que venir aquí le ayudará a dejar de ser infeliz?
Usuario:	Necesito ayuda, de eso no hay duda.
ELIZA:	¿Qué supondría para usted conseguir ayuda?
Usuario:	Quizá aprendería a soportar a mi madre.
ELIZA:	Cuénteme algo más acerca de su familia.

Uno de los programas de inteligencia artificial más conocido es el sistema ELIZA (Weizenbaum, 1966). Su objetivo es emular el comportamiento de un terapeuta rogeriano. La información requerida por el programa para llevar a cabo dicha tarea es codificada por un conjunto de reglas simples de asociación entre plantillas y patrones. El programa funciona, básicamente, emparejando las plantillas con una oración del usuario para encontrar los patrones apropiados para generar una respuesta.

Este método de generación de oraciones puede ser calificado, de forma algo inexacta, como un sistema de ranura y relleno. El conocimiento contenido en este tipo de generadores consiste en un repertorio extenso de frases y oraciones preparadas *ad hoc* con ranuras vacías que serán rellenas durante su ejecución por argumentos de la entrada recibida u otras plantillas aplicadas en el proceso.

Las limitaciones de esta técnica son obvias. Sin embargo, en dominios de discurso restringido, produce resultados muy interesantes. Actualmente, este procedimiento está integrado en algunas aproximaciones lexicalistas de la representación de información sintáctica (véase Danlos, 1983, 1987; Jacobs, 1985, 1987; Hovy, 1988).

En las gramáticas léxico-sintagmáticas la información formativa contenida en las reglas gramaticales está incorporada en la representación del lexicon. Con esto, se pretende homogeneizar la representación y el acceso de las propiedades formativas de las palabras individuales, de las expresiones idiomáticas y de las reglas gramaticales. Para ello, cuenta con dos tipos de estrategias alternativas. Por un lado, la técnica de las plantillas es utilizada para representar las estructuras constitutivas de unidades morfológicas y sintácticas que no contienen información procedimental. Por otro lado, cuando se requiere de acciones adicionales que modifican el estado del proceso de generación, la estrategia a seguir es la de definición de procedimientos.

Así, por ejemplo, para la siguiente representación semántica:

EVENT1 =: BARGAIN	HUM1 =: PERSON	HUM2 =: PERSON	TOK1 =: VEHICLE
SELLER = HUM1	NAME : "Juan"	NAME : "María"	TYPE : bicycle
BUYER = HUM2	SEX : masc	SEX : fem	NUMBER : 1
MERCHANDISE = TOK1			

el sistema de generación cuenta con patrones:

?BUYER *comprar* ?MERCHANDISE *a* ?SELLER

y plantillas:

```
[C1] = (:C1 [subject] [verb] eltn(0≤n≤2))
[subject] = (:subject token)
[verb] = (:verb verb)
elt = object / [attribute]
object = [dir-object] / [prep-object]
[dir-object] = (:dir-object token)
[prep-object] = (:prep-object [prep] (:object token))
[prep] = (:prep preposition)
```

para generar la siguiente plantilla:

```
(:C1 (:subject HUM2) (:verb buy) (:dir-object TOK1) (:prep-object (:prep from) (:object HUM1)))
```

equivalente a:

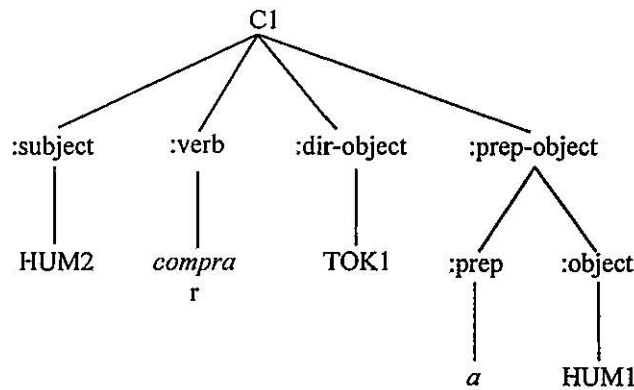


Figura 3.8.— Representación gráfica derivada de la plantilla generada en el ejemplo.

Uno de los generadores tácticos que expondremos posteriormente en este capítulo, el sistema de realización sintáctica del sistema PAULINE (Hovy, 1988), es un ejemplo de generadores construidos en base a las especificaciones de esta clase de gramática desarrollada desde la perspectiva de la lingüística computacional.

3.3.6. Gramática sistémica funcional

Desarrollada inicialmente por Halliday (1961, 1978, 1985), se caracteriza por su énfasis sobre los aspectos funcionales del lenguaje. Esta perspectiva funcional de la teoría lingüística tiene como su principal objeto de estudio la relación entre las funciones del lenguaje y la

naturaleza del sistema lingüístico. Esto es, analizar las estructuras lingüísticas en términos de sus propiedades funcionales.

Desde la lingüística sistémica funcional, la gramática se define, por tanto, como el instrumento que permite la articulación de los distintos aspectos funcionales del lenguaje para su integración en un formato de estructura lingüística. En ella se describen todas las estructuras gramaticales apropiadas para un lenguaje específico como un conjunto de decisiones independientes codificadas en redes discriminativas llamadas sistemas. Estos son listados de alternativas en los que están representados las opciones que un hablante dispone para construir palabras, sintagmas o cláusulas. Por ejemplo:

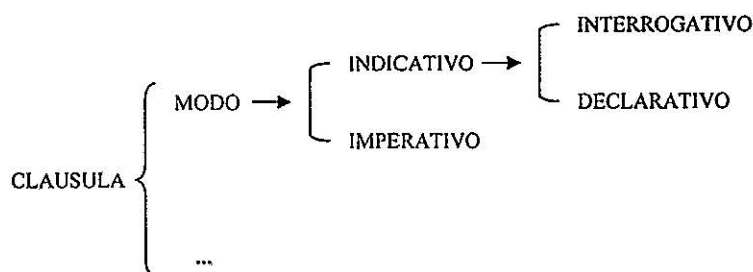


Figura 3.9.—Red discriminativa para el sistema MODO de la unidad lingüística cláusula.

En esto observamos que la definición de la unidad lingüística cláusula contiene un sistema MODO en el que están representadas las opciones que el usuario dispone para seleccionar el modo de la expresión lingüística. En este ejemplo, se han aplicado dos de los sistemas básicos:

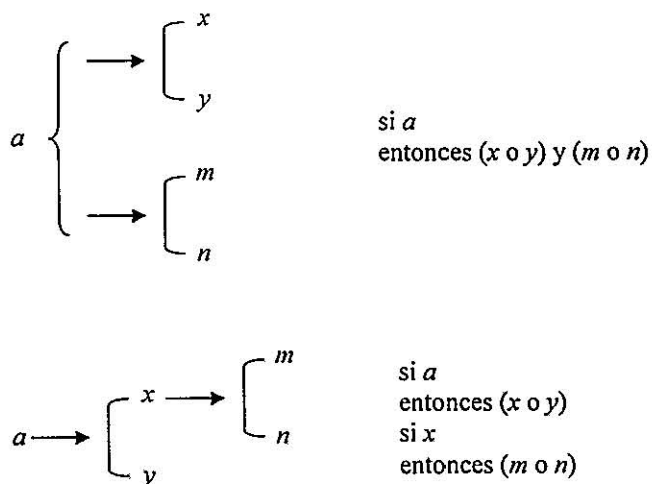


Figura 3.10.—Sistemas lógicos básicos utilizados en el ejemplo anterior.

Estas redes de sistemas son, asimismo, la herramienta básica de esta gramática para la expresión de las relaciones funcionales del lenguaje. Halliday distingue en el uso del lenguaje tres componentes funcionales o macrofunciones: la función ideativa, la función interpersonal y la función textual.

Una de las funciones fundamentales del lenguaje es la expresión de un contenido semántico en función de la experiencia del hablante sobre el mundo real, esto es, la *función ideativa*. Así, una unidad estructural como la cláusula, es la herramienta lingüística por medio del cual expresamos nuestra experiencia acerca de los procesos (acciones, procesos mentales o relaciones) acompañados por sus participantes (*e.g.*, agente, meta, experimentante) y sus circunstancias (*e.g.*, locación, instrumento). Este componente funcional está representado por el sistema de TRANSITIVIDAD. De modo simplificado, podemos distinguir en él los procesos mentales de los procesos materiales. Estos últimos cuentan, a su vez, con rasgos gramaticales adicionales.

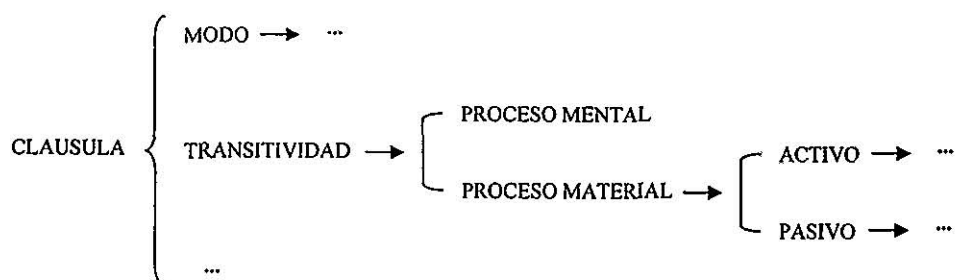


Figura 3.11.—Red discriminativa para el sistema de TRANSITIVIDAD de la unidad lingüística cláusula.

Además de transmitir información, el lenguaje sirve para establecer y mantener relaciones sociales. Esta función es definida como la *función interpersonal* y hace referencia al propósito de la conducta lingüística. Esto es, el lenguaje como medio para expresar creencias, opiniones, dudas, sentimientos personales, para hacer preguntas o para responder a ellas. Este componente funcional del sistema gramatical está contemplado, a nivel clausal, por el sistema MODO que hemos expuesto en un ejemplo anterior.

Asimismo, el uso del lenguaje en situaciones reales no es una simple referencia a una gramática y a un diccionario, sino que debe ser adecuado al contexto del discurso. Cuestiones como el TEMA, es decir, la selección del elemento inicial de la cláusula, o la distinción entre la

información dada y nueva deben ser consideradas en la construcción de los textos. Esta necesidad de reflejar la coherencia y la cohesión en el uso del lenguaje es definida como la *función textual*.

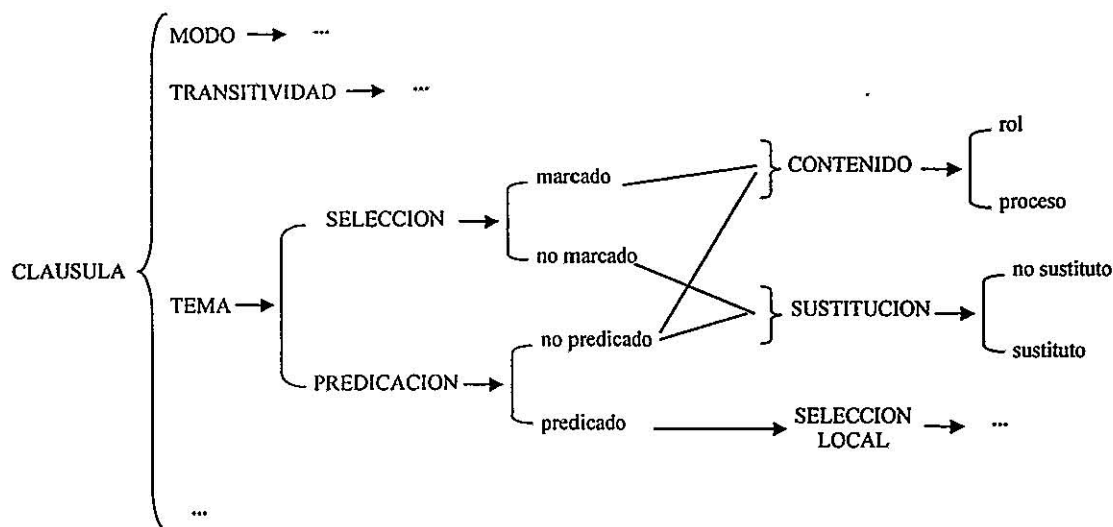


Figura 3.12.—Red discriminativa para el sistema TEMA de la unidad lingüística cláusula.

En este ejemplo, se ha introducido un nuevo tipo de sistema:

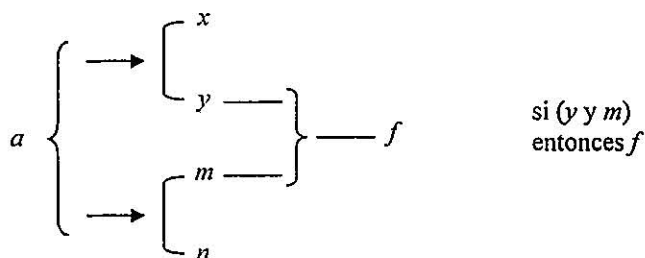


Figura 3.13.—Sistema lógico básico utilizado en el ejemplo anterior.

Hemos visto hasta el momento algunas opciones que definen la unidad lingüística cláusula. Otras unidades lingüísticas son consideradas en esta gramática: oraciones, sintagmas, palabras y morfemas. Las unidades están organizadas en un formato jerárquico que les dota de dos propiedades, la estructura y la clasificación. La estructura de las unidades se define por las unidades de rango inferior que la componen. Cada una de estas cumple una función particular: sujeto, predicado, complemento, adjunto. La clasificación de las unidades es determinada por las funciones del rango superior: verbal-predicado, nominal-sujeto/complemento, adverbial/adjunto.

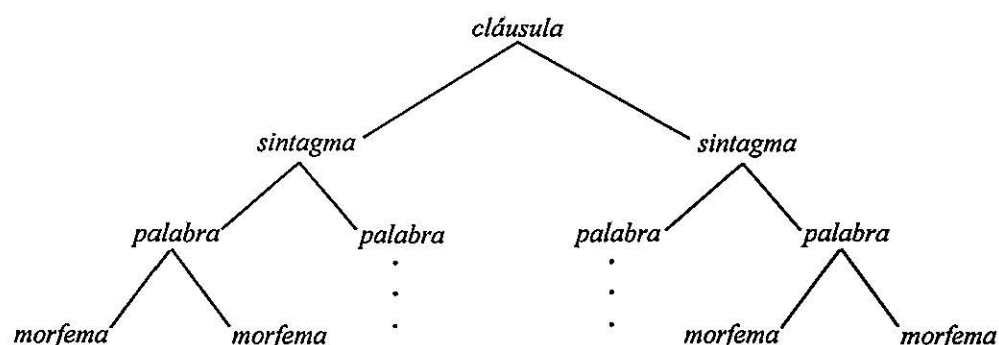


Figura 3.14.—Representación de la estructura jerárquica de las unidades lingüísticas.

Por último, nos queda hacer referencia a las reglas de realización. Estas son el mecanismo que permite la integración entre las categorías funcionales y las categorías estructurales. Cada sistema, por medio de sus características y funciones, impone alguna restricción al formato final de la oración. Las reglas de realización son el mecanismo para relacionar y manipular estas restricciones. Se han definido cinco tipos de reglas de realización.

Tipo de regla	Acción	Operador
Inclusión	Incluye al rasgo todas las funciones	+
Asignación	Asigna a la función otra función	=
Clasificación	Clasifica el rasgo con una función	/
Ordenación	Establece el orden de las funciones	>
Comprobación	Comprueba la no contradicción entre las funciones	//

Tabla 3.1.—Tipos de reglas de realización.

Así, por ejemplo, en la red sistémica representada (ver Figura 3.15.) contiene reglas que indican que todas las cláusulas deben tener un verbo con función predicativa (inclusión), que las cláusulas declarativas deben llevar un sujeto que precede al predicado (ordenación), que el elemento finito será un verbo conjugado (clasificación) cuyo número y persona deben concordar con los del sujeto (comprobación).

Con esto, finalizamos nuestra breve exposición de los conceptos básicos que definen las gramáticas sistémicas funcionales. Son fundamentalmente teorías lingüísticas. Se han realizado distintas aproximaciones a su aplicación computacional, siendo la de mayor relevancia el

sistema gramatical NIGEL (Mann y Matthiessen, 1983, 1985) en el que se ha fundamentado uno de los generadores tácticos más importante de la actualidad: el sistema PENMAN.

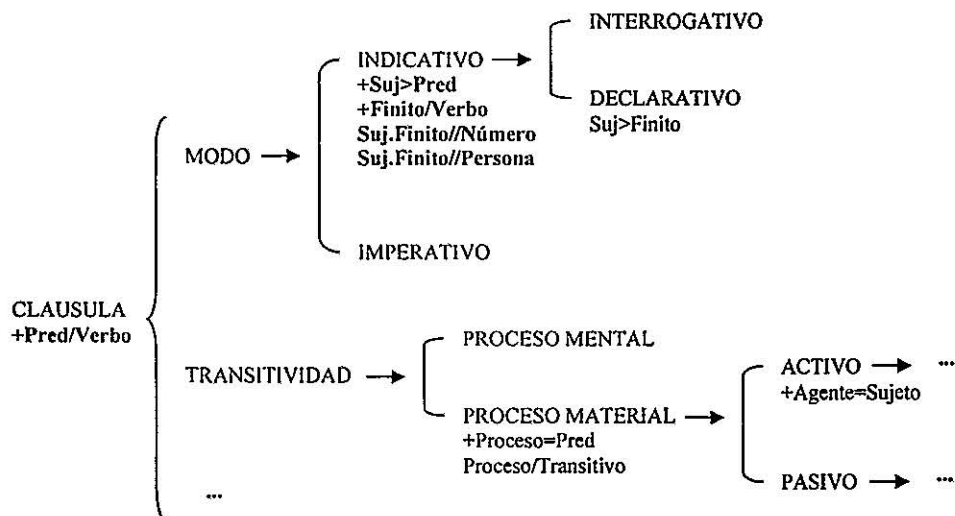


Figura 3.15.— Un ejemplo de parte de una red sistémica correspondiente a la unidad lingüística cláusula.

3.4. Sistemas de generación táctica

A continuación vamos a exponer con cierto detalle el funcionamiento de algunos generadores tácticos. En primer lugar, vamos a presentar el sistema BABEL de Goldman (1975). Dos de sus características le diferencian de otros sistemas de la primera generación. Por un lado, la aplicación de las redes de transición aumentada, utilizada preferentemente para sistemas de análisis, al proceso de generación. Por otro lado, la utilización de una representación conceptual como la entrada inicial del sistema. En segundo lugar, haremos referencia al generador táctico del sistema COMET (McKeown *et al.*, 1990). El interés de este sistema radica en la utilización de la metodología de la unificación, técnica que impera actualmente en el ámbito del procesamiento de lenguaje natural. Los dos sistemas restantes que presentaremos, PAULINE (Hovy, 1988) y PENMAN (Mann y Matthiesen, 1985; PENMAN, 1989), son fundamentales para el desarrollo del presente trabajo, ya que aspectos específicos de ambos sistemas han sido incorporados en el diseño de nuestro propio sistema de generación de oraciones.

3.4.1. MARGIE-BABEL

MARGIE (Meaning Analysis, Response Generation and Inference in English) fue construido por Roger Schank y su equipo (Schank, 1975) con el objeto de examinar su teoría de representación semántica, la teoría de dependencia conceptual. Cuenta con la participación de tres módulos, cada uno dedicado a una función específica. Estos son: el analizador conceptual, el realizador de inferencias y el generador de textos.

El módulo de generación de textos fue desarrollado por Neil Goldman (1975). Recibe como entrada una representación semántica definida en términos de las primitivas semánticas descritas en la teoría de dependencia conceptual. Esta representación, producida por el analizador conceptual o por el realizador de inferencias, es convertida en texto por el sistema BABEL.

Para ello, en primera instancia, el sistema selecciona las palabras específicas para denominar los elementos conceptuales representados en la entrada. Esto se lleva a cabo mediante una red discriminativa (un árbol de decisión binaria). Este proceso se inicia con la selección del verbo apropiado para expresar el evento especificado por el esquema de dependencia conceptual. Con esto, el sistema recupera el marco asociado al verbo seleccionado donde están las especificaciones acerca de la estructura sintáctica y de la localización de la información requerida en el esquema conceptual. Sobre este marco opera un conjunto de funciones lingüísticas para precisar aspectos sintácticos como el modo y el tiempo verbal. Sobre esta representación sintáctica opera una red de transición aumentada para generar la estructura superficial.

3.4.2. COMET

COMET (CoOrdinated Multimedia Explanation Testbed) fue desarrollado para un sistema de mantenimiento y reparación de equipos (sistema de radio receptor-transmisor portátiles de la armada estadounidense). Su cometido es producir explicaciones sobre el mantenimiento y reparación de equipos integrando, de forma coordinada, información textual e información gráfica. Después de una breve descripción del sistema en su conjunto nos centraremos en su módulo de generación de texto (McKeown *et al.*, 1990).

El sistema entra en funcionamiento cuando recibe una petición de explicación. La primera tarea la realiza el planificador de contenido determinando cuáles de las informaciones representadas en las fuentes de conocimiento del sistema deben ser incluidas en la explicación. Con ellas construye una jerarquía de formas lógicas que el coordinador de medios recibe como su entrada. En esta fase del proceso se decide qué parte de la información será producida por el generador de textos y qué parte por el generador de gráficos. Para ello añade a las formas lógicas indicaciones adicionales. Esta jerarquía de formas lógicas anotadas es la entrada para los generadores de textos y gráficos, produciendo texto y gráfico respectivamente. Finalmente, el administrador de disposición final combina la salida de ambos generadores dándole un formato conjunto para su presentación.

El generador de texto, construido en base a las especificaciones de la gramática de unificación funcional, recibe una estructura de formas lógicas como entrada. En una primera fase, el seleccionador léxico enriquece la entrada con rasgos sintácticos especificados en los items léxicos. Esto conforma una descripción de rasgos totalmente lexicalizado y estructurado en términos de roles semánticos, con indicaciones adicionales de algunas restricciones sintácticas impuestas por los items léxicos.

Sobre esta descripción, el realizador superficial añade, por la unificación de la entrada con la gramática, información gramatical. Esto incluye tanto la definición de roles sintácticos para cada uno de los roles semánticos como el orden de expresión de los constituyentes sintácticos, esto es, su estructura sintáctica.

El proceso de generación en COMET es en primer lugar la selección de un conjunto de alternativas funcionales disponibles para un contexto particular. Solo después de esto, y en segunda instancia, procede a tomar decisiones sobre la estructura sintáctica superficial. Esto resalta su carácter funcional.

3.4.3. PAULINE

PAULINE (Hovy, 1988) es un programa de generación de textos que incorpora explícitamente la información pragmática en su ejecución: características del oyente, situación conversacional y metas interpersonales del hablante. Se caracteriza, a su vez, por su visión de la generación como un proceso "dirigido por metas", es decir, el control del proceso lo ejercen las metas que el sistema debe satisfacer en cada momento para cumplir con su objetivo.

Así, el proceso de generación se define como un proceso de evaluación de metas almacenadas en la estructura de datos central del sistema. Esta estructura de datos tiene la propiedad particular de las listas LIFO (last in, first out), es decir, el último elemento introducido en la lista es el primer elemento a extraer de ella. Esta lista, en el caso del sistema al que nos referimos, admite únicamente dos tipos de datos: palabras y metas (temáticas o sintácticas). Si el elemento a evaluar es una palabra, el programa genera una salida y no introduce ningún elemento nuevo en la lista. En el caso de que el elemento a evaluar es una meta, el programa transfiere el control de ejecución al procedimiento asociado a ella que, además de realizar las posibles acciones definidas, introduce nuevas unidades a la lista. El proceso de generación finaliza cuando la lista no contiene ningún elemento a evaluar.

El control de la generación táctica lo ejercen, por tanto, las metas sintácticas. Cada una de ellas cuenta con una función (el especialista sintáctico) que crea el ámbito sintáctico adecuado para alcanzar su meta. Esto es, el especialista sintáctico recibe una pieza de la representación inicial y, según el caso, alguna información adicional, y produce un listado de metas sintácticas y/o palabras. Así, sucesivamente y en forma de cascada, las metas sintácticas se van expandiendo en otras metas sintácticas y/o palabras hasta que todas las palabras sean generadas.

Los especialistas sintácticos alcanzan las metas creando un ámbito sintáctico para la entrada recibida. Para ello, deben conocer las distintas formas posibles y saber seleccionar la alternativa apropiada.

Dependiendo del tipo de meta sintáctica, los especialistas pueden ser muy simples o complejos. Los primeros que no contienen información procedimental son implementados, en PAULINE, como patrones. Estos son estructuras gramaticales representadas en el sistema que no prescriben acciones que el generador deba ejecutar. En caso contrario, cuando alcanzar la meta requiere acciones que procesan y alteran el estado del generador, el especialista es implementado como un procedimiento.

El proceso de generación táctica es, en este sistema, la aplicación de especialistas sintácticos. A diferencia de otros métodos de generación, los especialistas sintácticos no se activan en función de criterios predeterminados sino, como un proceso "dirigido por metas", son activados por la secuencia determinada por su predecesor. Vamos a ver con un ejemplo el

funcionamiento de este sistema de realización lingüística. Empezaremos con la siguiente meta sintáctica:

```
#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-SENT-TOP]
  [TOPIC : PTRANS-1]}
```

donde PTRANS-1 representa un caso particular de la primitiva PTRANS que significa, según la teoría de dependencia conceptual de Schank (1975), el cambio de localización de algún objeto físico.

```
#{ACTION = PTRANS-1
  [ACT : PTRANS]
  [ACTOR : John]
  [TO : New York]
  [TIME : PAST]}
```

El especialista SAY-SENT-TOP determina el tipo de oración a realizar. Para nuestro ejemplo, la nueva meta sintáctica construida por esta función es:

```
#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-EVENT-SENT]
  [TOPIC : PTRANS-1]}
```

Este especialista se expande en la secuencia

```
[SAY-PRE-SENT SAY-SUBJECT SAY-PREDICATE]
```

después de decidir el aspecto de la entrada que hará de sujeto de la oración y, si la interpretación de la entrada así lo requiere, seleccionar el verbo. Finalizado este procedimiento, en la estructura de datos central están almacenadas secuencialmente, en nuestro caso particular, las siguientes metas sintácticas:

```
#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-SUBJECT]
  [TOPIC : John]}
#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-PREDICATE]
  [TOPIC : PTRANS-1]}
```

SAY-SUBJECT crea una nueva meta sintáctica que es incorporada a la estructura de datos central, quedando ésta de la siguiente manera:

```

#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-NOUN-GROUP]
  [TOPIC : John]}
#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-PREDICATE]
  [TOPIC : PTRANS-1]}

```

El proceso continúa con la aplicación del especialista SAY-NOUN-GROUP hasta que todos los elementos expandidos a partir de él sean evaluados y el único elemento existente en la estructura de dato central sea la meta sintáctica

```

#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-PREDICATE]
  [TOPIC : PTRANS-1]}

```

La evaluación de esta meta da lugar a una nueva secuencia de metas:

```

#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-VERB]
  [TOPIC : PTRANS-1]
  [WORD : go]}
#{SYNTAX-GOAL
  [SAY-FUNCTION : SAY-PREPGROUP]
  [TOPIC : New York]
  [PREPOSITION : to]}

```

que serán expandidas sucesivamente, hasta que todos los elementos expandidos a partir de ellas sean evaluados.

3.4.4. PENMAN

El sistema PENMAN es uno de los sistemas de generación de oraciones individuales en inglés más amplio del mundo. Es un programa de generación táctica de propósito general desarrollado inicialmente por Mann y Matthiesen (1985) y posteriormente por investigadores de USC/ISI (1989), entre otros, Eduard Hovy. La importancia de este sistema se observa en su utilización como componente de realización lingüística en sistemas de traducción mecánica como JAPANGLOS y SPANGLOSS.

En PENMAN, el proceso de generación se realiza en dos fases. En primer lugar, la interpretación gramatical de la información semántica contenida en la entrada. Para ello, el generador deberá recorrer las redes sistémicas definidas por el sistema gramatical NIGEL

guiado por la información, no sólo ideativa, sino también interpersonal y textual, contenida en la entrada.

NIGEL define la gramática inglesa en términos de la teoría sistémica funcional. Así, los nodos de la red, los sistemas, representan las distintas opciones gramaticales. En cada uno de ellos, el generador selecciona un valor para el rasgo gramatical en cuestión, invocando el procedimiento de decisión asociado. Este proceso continúa hasta que queden ensamblados suficientes rasgos gramaticales para la especificación completa de una oración. Entonces el sistema entra en su segunda fase.

La tarea de generador consiste, ahora, en construir un árbol sintáctico y seleccionar las palabras que satisfagan los rasgos gramaticales seleccionados para finalmente generar la oración. Este proceso de interpretación estructural de los rasgos gramaticales se realiza aplicando las reglas de realización asociadas a cada uno de los rasgos.

Dada su repercusión para nuestro trabajo, vamos a exponer con más detalle algunos aspectos de la representación de información de este sistema.

Como ya hemos comentado, varios tipos de información deben estar disponibles para que PENMAN proceda a la generación de una oración: información conceptual, interpersonal y textual. Todo ello debe estar especificado en el formato SPL (Sentence Plan Language) que sirve de interfaz de entrada al sistema.

El SPL es una notación flexible que permite la integración de información de distintas clases. Vamos a verlo con un ejemplo:

```
(SI / SAIL
: actor (KNOX / SHIP
: name "Knox")
: destination (PEARL-HARBOR / PORT
: name "Pearl-Harbor")
: tense PRESENT
: speechact ASSERTION)
```

Este plan contiene información que el sistema necesita para generar la siguiente oración:

Knox sails to Pearl-Harbor.

SI, *KNOX* y *PEARL-HARBOR* son instanciaciones de los conceptos **SAIL**, **SHIP** y **PORT**, respectivamente. Estos suelen ir acompañados por informaciones adicionales que juegan roles específicos en su definición: *actor*, *destination*, *tense*, *speechact*, *name*.

En el ejemplo, la información de los roles *actor* y *destination* son instanciaciones de conceptos, *tense* y *speechact* contienen símbolos predefinidos, PRESENT y ASSERTION, y tenemos cadenas de caracteres en el rol *name* de los conceptos **SHIP** y **PORT**.

Otros muchos roles pueden ser definidos. Por ejemplo:

```
(SI / SAIL
: actor (KNOX / SHIP
: determiner THE
: number PLURAL)
: destination (PEARL-HARBOR / PORT
: name "Pearl-Harbor")
: time (T1 / DAY
: name TODAY)
: tense PRESENT
: speechact ASSERTION)
```

Se ha incluido para el concepto **SAIL** el rol *time*, y para el concepto **SHIP** los roles *determiner* y *number*. Esto hace que el generador introduzca un complemento circunstancial y exprese el sujeto con un grupo nominal:

The ships sail to Pearl-Harbor today.

Es obvio decir que las entradas deben ser inteligibles para PENMAN, es decir, el sistema debe saber el significado de la información contenida en ellas. Por un lado tenemos símbolos predefinidos, como son THE, PLURAL, TODAY, PRESENT y ASSERTION, que el generador interpreta por medio de macros¹ definidas en el programa. Por otro lado están los conceptos, en el ejemplo, **SAIL**, **SHIP**, **PORT** y **DAY**. Estos deben formar parte del conjunto de entidades del mundo que PENMAN conoce.

PENMAN organiza las entidades del mundo en una taxonomía. Esto se debe a que hay formas distintas de referirse con el lenguaje a diferentes tipos de entidades que aparecen en el mundo. Así, las categorías definidas en esta taxonomía reflejan distinciones gramaticales. Por ejemplo, en un primer nivel, las entidades son clasificadas en tres clases: *object*, *quality* y

¹ Las macros son conjuntos de acciones agrupadas en un solo comando que se realizan automáticamente.

process; por lo general, su realización lingüística corresponde, respectivamente, a grupos nominales, adjetivales y verbales.

Esta taxonomía consta de dos partes: el *Upper Model* y el *Domain Model*. Las entidades del *Upper Model* (Bateman, Kasper, Moore, y Whitney, 1990) crean una partición muy abstracta del mundo y forman parte intrínseca del sistema. El *Domain Model* contiene la definición de las entidades que aparecen en un dominio particular. Todas ellas deben estar subordinadas al *Upper Model* y ser definidas por el usuario.

Como ejemplo, **SAIL** es un concepto definido en nuestro *Domain Model*. Es una especialización de la categoría *Motion-Process* del *Upper Model* que a su vez es una especialización de *Nondirected-Action*. Vamos a ver gráficamente la relación entre **SAIL** y el *Upper Model*:

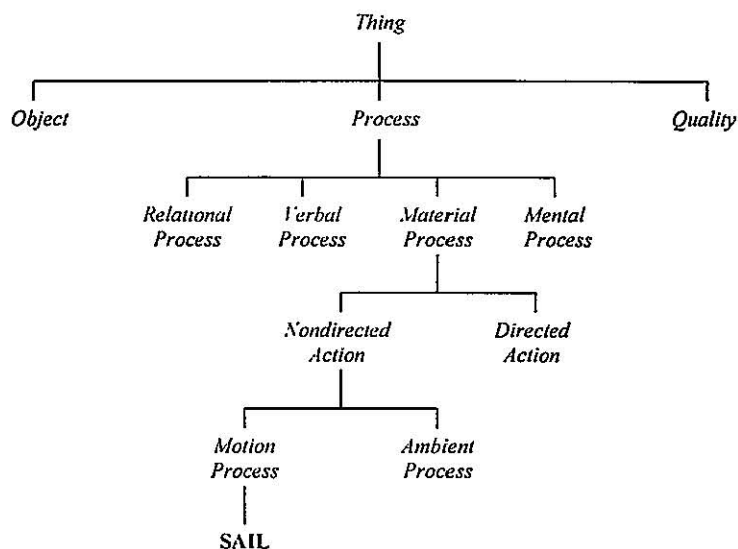


Figura 3.16.—Upper model para el concepto SAIL.

Finalmente, todos los conceptos del *Domain Model* deben tener asociados elementos léxicos definidos, por el usuario, en el lexicon. En PENMAN, el lexicon no es más que una lista de palabras con sus variantes (formas verbales, plurales, etc.) y algunas otras características. El propio sistema proporciona un conjunto de palabras al lexicon, palabras de función gramatical como preposiciones y verbos auxiliares.

3.5. Recapitulación

En este capítulo hemos presentado las principales herramientas que la inteligencia artificial ha desarrollado para la implementación computacional de los procesos de codificación gramatical. Muchos de estos aspectos son esenciales para la realización de este trabajo. Sin embargo, debemos realizar algunas consideraciones generales acerca de las implicaciones que para la psicología tienen estas técnicas.

Si bien cualquiera de ellas tiene una importante relevancia para la inteligencia artificial, desde la psicología se han establecido una serie de restricciones que han de cumplir los sistemas que pretendan simular los procesos de codificación gramatical. Por esto, con respecto a las técnicas que hemos abordado, hemos de señalar que no todas ellas cumplen tales criterios, aunque no por ello podíamos obviarlas en este capítulo. No obstante, por su interés en el tema que nos ocupa prestaremos nuestra atención a las más actuales.

Con respecto a la gramática léxico-sintagmática, implementada en el sistema PAULINE diremos que es la que hasta ahora más eficazmente ha resuelto algunas de las dificultades planteadas en los intentos de modelado computacional de los procesos de codificación gramatical (ver Capítulo 2, página 48). No ha ocurrido lo mismo, sin embargo, con las implementaciones de las técnicas de unificación y de la gramática sistémica funcional, de COMET y PENMAN respectivamente, que al igual que el modelo de Schlesinger no respeta la naturaleza incremental de los procesos de producción verbal. Asimismo, si bien el sistema COMET se mostraba como altamente eficiente, no resultaba del todo adecuado para el análisis psicológico al no permitir un seguimiento detallado de las operaciones que iba realizando en cada momento. Con respecto al PENMAN, señalaremos su estructura de representación conceptual, ya que permite integrar distintas clases de información y se adapta adecuadamente a las exigencias de los procesos de generación táctica.

Hasta el momento hemos expuesto la información más relevante en la que hemos basado nuestro estudio acerca de los procesos de codificación gramatical y que ha dado lugar al sistema de generación de oraciones GEDEON que presentaremos a partir de este punto.

Capítulo 4: Sistema de generación de oraciones GEDEON

4.1. Introducción

GEDEON (GEnerador DE Oraciones en lenguaje Natural) es un programa informático de producción de oraciones individuales en castellano para un dominio conceptual limitado (consultar Anexo para acceder a información detallada sobre el sistema). Es, por tanto, un caso particular de generador táctico de lenguaje natural que tiene el objetivo de construir estructuras lingüísticas a partir de una entrada de naturaleza conceptual de estructura no lineal. Para ello, el sistema requiere de información acerca de la gramática y el léxico de la lengua castellana así como funciones para la selección y ordenación de los elementos lingüísticos. Este capítulo está dedicado a la descripción de los componentes del sistema GEDEON y el procedimiento de desarrollo aplicado en su construcción. Pero antes de pasar a ello, nos detendremos brevemente en algunas cuestiones de carácter general.

En primer lugar nos referiremos a los supuestos básicos asumidos en el diseño del sistema de generación táctica. En primer lugar, en cuanto a la entrada inicial del sistema, consideramos que ésta debe ser una estructura representacional de naturaleza fundamentalmente semántica e independiente de las propiedades sintácticas particulares de los lenguajes naturales. Estas estructuras de representación son producidas por un sistema modular (el conceptualizador o el generador estratégico, para la psicolingüística y la inteligencia artificial, respectivamente) y forman parte de una red jerárquica con capacidad de heredar propiedades definidas por los elementos de niveles superiores.

Asimismo, el proceso de generación es considerado como un proceso "dirigido por metas", esto es, controlado por objetivos. Dada una meta, el sistema debe realizar las acciones definidas para alcanzarla. Estas acciones pueden, a su vez, crear nuevas metas que descomponen la original en un conjunto de submetas que permite su realización. Este procedimiento tiene elementos comunes a las estrategias utilizadas en los sistemas de producción guiados por objetivos. Sin embargo, en la generación de lenguaje natural, el orden del tratamiento de los objetivos tiene un efecto directo sobre el orden de aparición de los elementos lingüísticos en el

texto de salida. Por esto, es necesario introducir en el proceso de generación el control explícito del orden de realización de las metas.

Una segunda cuestión de carácter general a considerar es la herramienta de desarrollo utilizada en la construcción del sistema. Las características de la programación orientada a objetos se muestran útiles, y en algunos casos imprescindibles, para abordar el problema de generación de oraciones. Esta metodología de programación no sólo resulta adecuada para la representación declarativa del conocimiento (las estructuras de ranura y relleno débil como redes semánticas y marcos), sino que permite la integración en dicha representación de aspectos procedimentales del conocimiento. Esta característica es especialmente relevante para la representación de los conocimientos implicados en los procesos objeto de este trabajo. Así, por ejemplo, la realización lingüística de una estructura de representación conceptual de una relación conjuntiva no es la misma que la de una relación causal, aún cuando ambas pertenezcan a la misma categoría sintáctica, la oración. En esto observamos la necesidad de definir funciones particulares, sensibles a la clase de estructura de representación conceptual en la generación de una oración. Las propiedades de encapsulamiento y de polimorfismo de la programación orientada a objetos (comentadas en el capítulo 1 de este trabajo) están encaminadas a resolver este tipo de cuestiones.

Asimismo, la capacidad de herencia, no sólo de los atributos sino también de las operaciones, hace de ésta una metodología especialmente adecuada para la representación de conocimientos lingüísticos. Por ejemplo, la clase *Verb*² tiene como atributos el modo, el tiempo, la persona y el número, comunes a todas las palabras del lexicon que pertenezcan a esta categoría gramatical. Cada una de éstas pertenece además a una clase específica de conjugación, por ejemplo, la primera conjugación regular, de terminación "ar", en la que está definida la operación de asignación de afijos. Esta operación, al igual que los atributos, es heredada por todas las clases que de ella se derivan y por tanto, aplicada para la asignación de afijos de todos los verbos que pertenezcan a dicha conjugación verbal.

Con esto hemos pretendido ejemplificar las posibilidades que la programación orientada a objetos ofrece para el modelado computacional de los procesos de generación de oraciones. Su potencia e idoneidad, así como sus limitaciones, se irán poniendo de manifiesto a lo largo de la exposición de este capítulo. Sin intención de eludir nuestra responsabilidad en la elección de la

² A lo largo de este capítulo, hemos respetado los términos utilizados en la programación del sistema para facilitar su comprensión.

herramienta de desarrollo, es importante resaltar que, actualmente, ésta es la metodología predominante en la investigación y desarrollo de las distintas áreas de la inteligencia artificial, tanto es así, que lenguajes de programación tradicionales como el LISP están siendo adaptados con la incorporación de los conceptos básicos de la programación orientada a objetos.

Para este trabajo hemos optado por un lenguaje de programación orientada a objetos de propósito general, el C++ (Borland C++ 5.01), por su versatilidad como lenguaje de programación y nuestra preferencia por los lenguajes compiladores frente a los intérpretes.

4.2. Descripción del sistema

Considérese la siguiente estructura de representación conceptual:

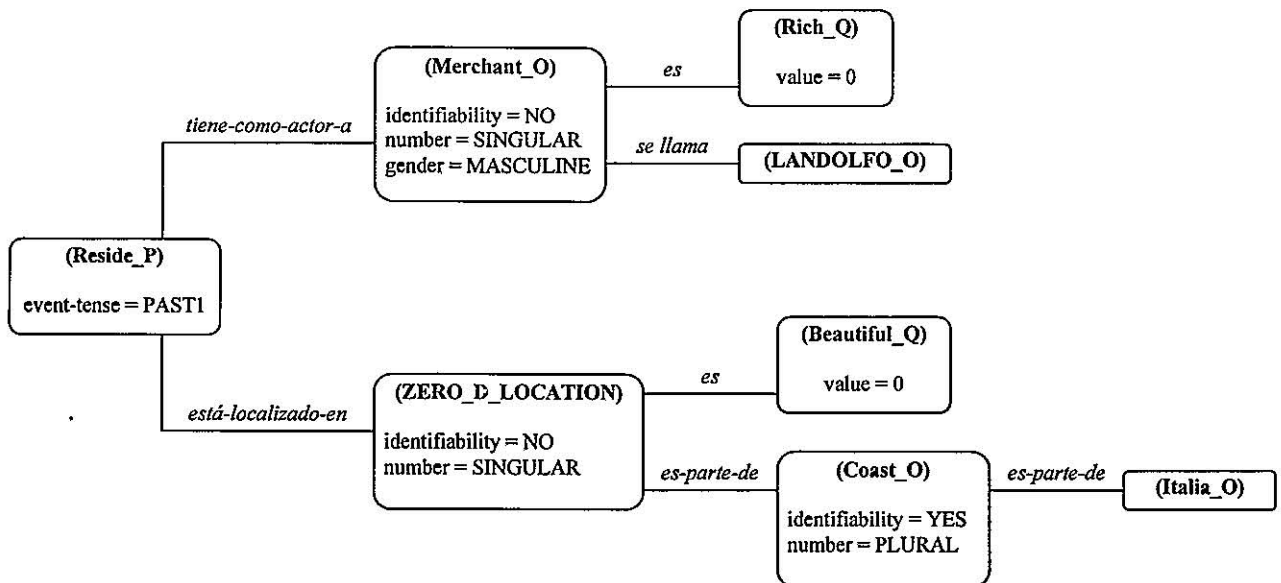


Figura 4.1.— Un ejemplo de representación conceptual.

A partir de esta representación, producir oraciones como "en un hermoso lugar de las costas de Italia, vivía un rico comerciante llamado Landolfo" o "un rico comerciante que se llamaba Landolfo vivía en un hermoso lugar de las costas de Italia" exige del sistema la generación de secuencias de metas que consideren elementos de la entrada progresivamente más pequeños hasta que finalmente el elemento a considerar pueda ser expresado por una palabra o frase del lexicon.

Esto es, el generador comienza su ejecución con la meta de construir una oración con una entrada X (en el ejemplo de la *Figura 4.1.*, Reside_P). De su evaluación se obtiene una secuencia de metas. Así, el próximo objetivo del generador podría ser la de construir el sujeto de la oración con el elemento de entrada X1 (Merchant_O) que a su vez será realizado por la estructura lingüística de sintagma nominal en la que debe considerar asimismo los elementos X11 (Rich_Q) y X12 (LANDOLFO_O) del elemento X1.

En este proceso de evaluación recursiva de metas intervienen conocimientos de tipo conceptual, sintáctico y léxico. La organización de estos conocimientos y su modo de participación en los procesos de generación de oraciones son cuestiones que trataremos a continuación.

4.2.1. Arquitectura del programa

Muchos son los conocimientos necesarios para la producción del lenguaje natural. En el caso particular de la realización lingüística, es imprescindible la participación de la gramática y el lexicon. Igualmente lo es la presencia de formatos representacionales de información conceptual adecuados para la caracterización de los "mensajes preverbiales". Todos ellos están integrados en el sistema GEDEON. Con esto no nos referimos a que el sistema contenga todas las palabras y expresiones perifrásticas de la lengua castellana ni todas las reglas gramaticales descritas por la lingüística. Como ya hemos mencionado con anterioridad, el sistema GEDEON es un generador de oraciones individuales para un dominio conceptual restringido. Sin embargo, en él están presentes todos los elementos necesarios y suficientes para la generación táctica de lenguaje natural. Esto es, la limitación del sistema es básicamente una cuestión cuantitativa y en ningún caso cualitativa, es decir, su capacidad para generar oraciones está limitada únicamente por la cantidad de conceptos, palabras y estructuras sintácticas implementadas en el programa.

Por tanto, el sistema GEDEON es un programa informático de generación táctica que, dada una estructura conceptual, produce una oración adecuada para la expresión lingüística de los contenidos semánticos representados en la entrada recibida. Para ello, cuenta con la intervención de diversos tipos de información y procedimientos, organizados de la siguiente forma:

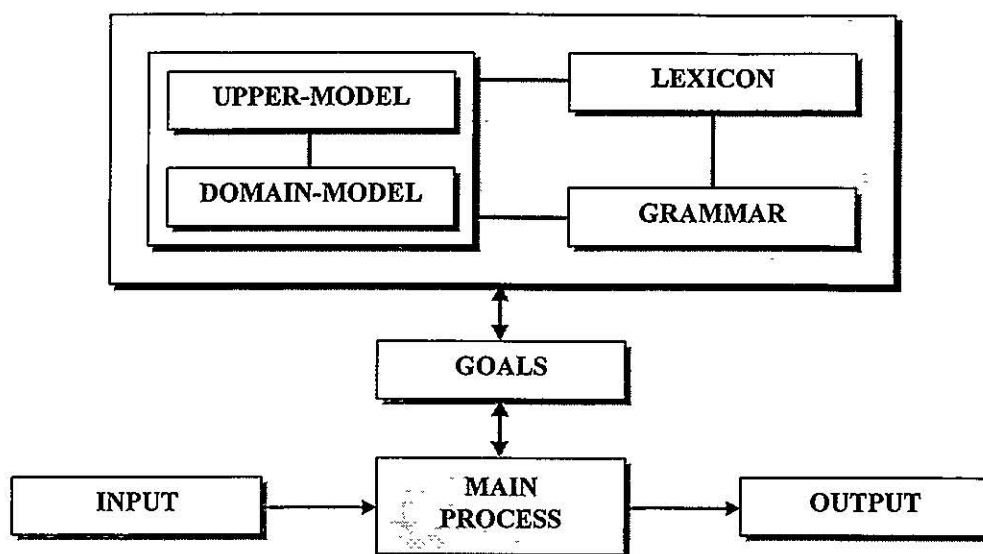


Figura 4.2.—Arquitectura del programa.

En cada uno de los componentes del sistema está recogida la información necesaria que le capacita para la realización de una función específica en el proceso de generación de oraciones. A continuación describiremos con detalle cada uno de estos componentes.

4.2.1.1. *Upper-Model*

Uno de los objetivos contemplados en el proyecto PENMAN (1989) es el de desarrollar un recurso computacional de representación del conocimiento conceptual apropiado para la realización lingüística. El resultado es un sistema de clasificación de entidades conceptuales independiente del dominio y de la tarea, el *upper-model* (Bateman, Kasper, Moore y Whitney, 1990).

Es una jerarquía de categorías muy generales y abstractas de entidades conceptuales del mundo en la que se reflejan, asimismo, sus diferencias en el formato de expresión lingüística. Así, los objetos son generalmente expresados mediante nombres mientras que las acciones son realizadas lingüísticamente a través de los verbos.

En este módulo del sistema GEDEON está implementado un subconjunto de las entidades conceptuales definidas en el modelo original, aquéllas que participan directamente en la representación conceptual de nuestro dominio particular.

La clase base de esta jerarquía es la clase *THING*. Es básicamente una clase abstracta que sirve para vincular todos los elementos representados en el *upper-model*. De ella se derivan directamente tres subclases: *PROCESS*, *OBJECT* y *QUALITY* que conforman sus tres subjerarquías principales.

4.2.1.1.1. La subjerarquía *PROCESS*

Todas las entidades clasificadas en esta subjerarquía pueden ser expresadas mediante un verbo. Cada clase de «proceso» es en sí una entidad conceptual en la que están definidas, como atributos, los participantes y las circunstancias asociadas al proceso en cuestión. Estas entidades están organizadas en el *upper-model* de la siguiente forma:

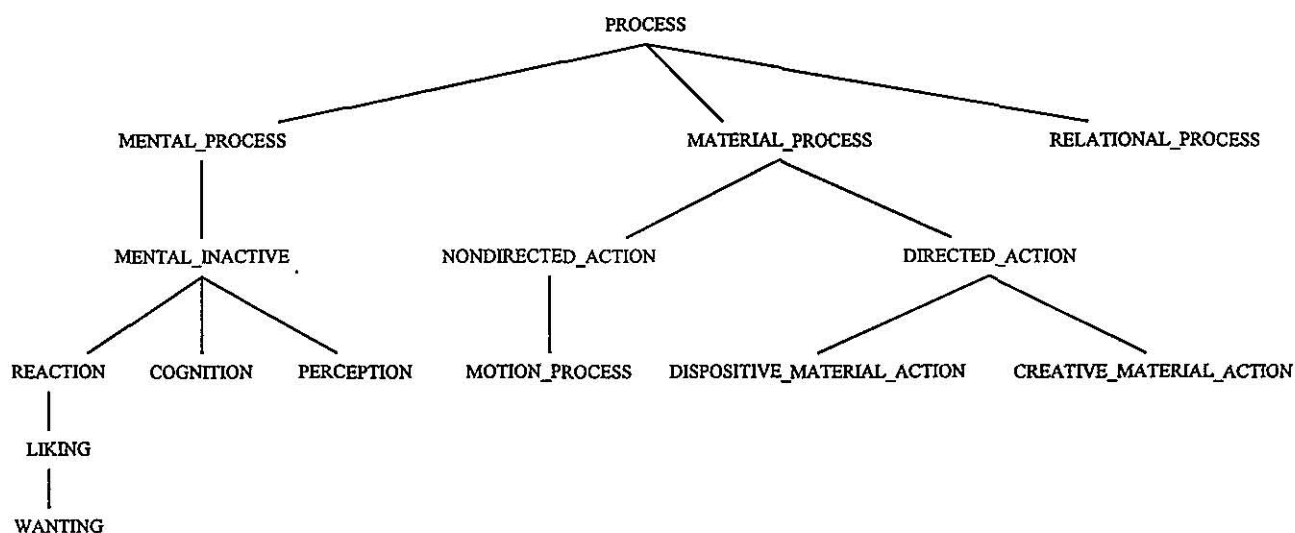


Figura 4.3.— Organización de las subclases de la clase *PROCESS* en el *upper-model*.

Una primera categoría de «procesos» son los *MENTAL_PROCESS*. A ella pertenecen las acciones internas de los individuos como "pensar", "sentir", "ver" o "decidir". Son considerados *MENTAL_INACTIVE* los conceptos que describen acciones mentales que no exigen un esfuerzo adicional por parte del actor (ejemplos de estos casos son acciones como "convencer" o "rogar"). Tres subclases son definidos en el *upper-model*: *PERCEPTION* (e.g., "ver", "oír", "sentir"), *COGNITION* (e.g., "pensar", "creer", "saber"), y *REACTION* (e.g., "querer", "gustar", "desear").

Los *MATERIAL_PROCESS*, por su parte, describen «procesos» que se producen en una dimensión física. Estas acciones pueden ser realizadas por agentes externos, en cuyo caso, son

consideradas entidades de la categoría *DIRECTED_ACTION*. Asimismo, el elemento sobre el cual se realiza este tipo de acciones pueden ser un objeto existente o no. De esta clasificación se obtienen dos especializaciones de la clase *DIRECTED_ACTION*, de las que se derivan «procesos» que actúan sobre objetos existentes (*DISPOSITIVE_MATERIAL_ACTION*) o que crean nuevos objetos (*CREATIVE_MATERIAL_ACTION*). Esta distinción refleja la diferencia que podemos observar entre, por ejemplo, "comer la tarta" y "hacer la tarta" (que corresponden a *DISPOSITIVE_MATERIAL_ACTION* y *CREATIVE_MATERIAL_ACTION*, respectivamente).

Otra clase derivada de *MATERIAL_PROCESS* es la clase *NONDIRECTED_ACTION*. Esta clase de acción no requiere la intervención de un agente causal externo. Una clase particular de este tipo de «procesos» es la clase *MOTION_PROCESS* en la que el concepto de acción representada incluye movimiento del «actor».

Acabamos de mencionar a uno de los participantes comunes a todas las entidades conceptuales descritos hasta el momento, el «actor». Su presencia es obligatoria para todas las categorías de *MENTAL_PROCESS* y *MATERIAL_PROCESS*, aún cuando su significado no sea idéntico para todos los casos. Esto es, mientras que para la clase *DIRECTED_ACTION*, el rol de «actor» representa el agente causal de la acción, para la clase *NONDIRECTED_ACTION* representa el medio por el cual se realizan los «procesos». De igual modo es definido otro participante común a las clases de «procesos» mencionados, el «actee». Este elemento es imprescindible para la definición de conceptos de la categoría *DIRECTED_ACTION* en donde el «actee» describe la entidad objeto de la acción. Sin embargo, para los «procesos» de *MENTAL_INACTIVE* y *NONDIRECTED_ACTION*, la presencia de un elemento que desempeñe el rol de «actee» es opcional. Es más, en el caso de *NONDIRECTED_ACTION*, más que un genuino participante de la acción, su misión es definir la naturaleza de la acción (e.g., "subir una montaña").

En la definición de un «proceso» intervienen, a su vez, diversas circunstancias asociadas. Algunas de ellas son comunes a todas las clases, como son «location», «time», o «manner». Otras, en cambio, son específicas para determinadas categorías de «procesos». En el caso concreto de la clase *MOTION_PROCESS*, existen dos circunstancias adicionales que potencialmente acompañan esta clase de acciones: «source» y «destination».

Sin entrar en detalles, podemos anotar en este punto de la exposición que el «proceso», sus participantes y las circunstancias asociadas, son realizados lingüísticamente por el verbo, el

sujeto, los objetos directo e indirecto y los complementos circunstanciales. Sin embargo, el formato lingüístico específico aplicado para cada caso estará determinado, por distintos aspectos del nivel conceptual, como las categorías de las que se deriva la entidad que acompaña, la definición del «proceso» en cuestión y la relación establecida entre ambas entidades conceptuales. Así, por ejemplo, si la entidad conceptual que actúa como participante «actee» de la acción es un elemento de la categoría *OBJECT*, la categoría sintáctica asignada para la realización del objeto directo sería un sintagma nominal. Si, en cambio, dicho elemento es una clase derivada de la categoría *PROCESS*, el formato lingüístico aplicado para su expresión sería el de una proposición subordinada sustantiva (e.g., "decide *la hora de salida*" o "decide *salir a una hora*"). Asimismo, las circunstancias «source» y «destination» asociadas con *MOTION_PROCESS*, son complementos circunstanciales de lugar que se expresan mediante la misma categoría sintáctica, el sintagma preposicional, en el que, sin embargo, reflejan su diferencia conceptual a través del enlace (e.g., "fue *desde casa*" o "fue *a casa*").

Pasamos, a continuación, a describir una clase especial derivada de *PROCESS*: *RELATIONAL_PROCESS*. A diferencia de otras, esta categoría describe la relación entre sus participantes en lugar de la acción de uno de los participantes sobre el otro. Las entidades conceptuales de esta categoría consideradas en nuestro *upper-model* están organizadas en la siguiente estructura jerárquica:

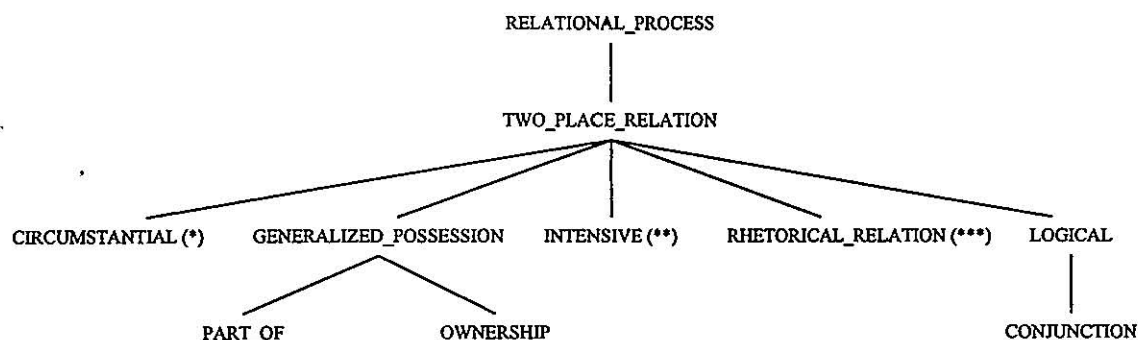


Figura 4.4.— Organización en el *upper-model* de la subjerarquía *RELATIONAL PROCESS* (Consultar Figuras 4.5., 4.6., 4.7.).

Todas las relaciones conceptuales aquí consideradas describen la relación entre dos participantes (*TWO_PLACE_RELATION*): «domain» y «range». Las clases que de ella se derivan son especializaciones de las relaciones que entre dos entidades conceptuales pueden ser establecidas.

GENERALIZED_POSSESSION describe la relación entre el poseedor («*domain*») y el poseído («*range*»). Cuando esta relación de posesión indica, a su vez, pertenencia, la entidad conceptual en cuestión es *OWNERSHIP*. Por su parte, la clase *PART_OF* es un caso particular de relación entre el todo y una parte en la que el «*domain*» es el todo y el «*range*» es una parte.

La clase *LOGICAL* es un «proceso» de composición de dos entidades conceptuales. Esta relación es conjuntiva para la clase derivada *CONJUNCTION*.

Otra clase derivada de *TWO_PLACE_RELATION* es la clase *CIRCUMSTANTIAL*. El caso más común de esta categoría es la relación entre un «proceso» y un «objeto» reflejada en los complementos circunstanciales, como por ejemplo, "*regresar a casa*" o "*vivir en la playa*". Sin embargo, esta clase de relación no se restringe únicamente a este tipo de casos, sino que las entidades participantes pueden ser ambas «objetos» o ambas «procesos».

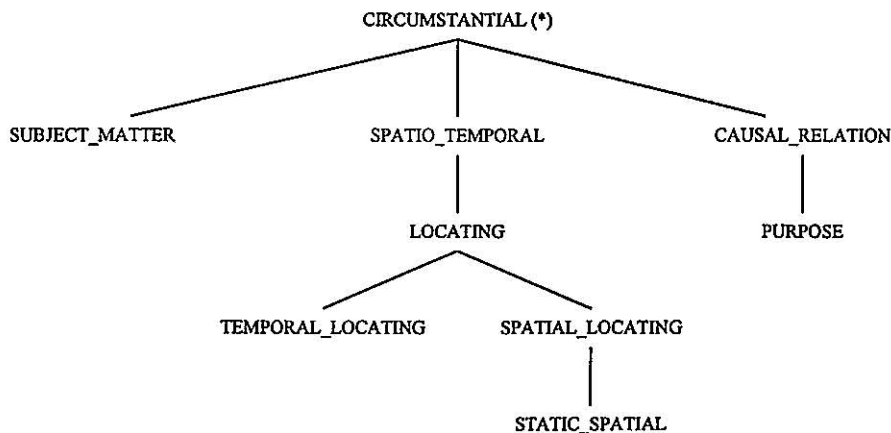


Figura 4.5.—Organización de las clases derivadas de *CIRCUMSTANTIAL* en el upper-model.

La clase *SUBJECT_MATTER* expresa relaciones del tipo «*domain*» "trata de" o "es acerca de" «*range*» (e.g., "este *trabajo* trata de la *codificación gramatical*"). *SPATIO_TEMPORAL* engloba las relaciones de dos entidades en el espacio o en el tiempo. Las clases que de ella se deriva consideradas en nuestro sistema son: *LOCATING*, en el que el «*range*» representa el espacio (*SPATIAL_LOCATING*) o el tiempo (*TEMPORAL_LOCATING*) en el que se sitúa una entidad conceptual («*domain*»). *STATIC_SPATIAL* es una especialización de la relación *SPATIAL_LOCATING* en la que no está presente ningún elemento de movimiento respecto a la localización especificada (e.g., "el *libro* está en la *biblioteca*").

CAUSAL_RELATION describe una relación entre la causa («*domain*») y el efecto («*range*»). Cuando esta relación expresa un efecto intencional en la que la entidad conceptual que desempeña el papel de la causa es considerada una acción futura respecto a la del efecto, la relación es categorizada como *PURPOSE* (e.g., "fue al supermercado para comprar la leche"). Esta relación es representada de forma similar por la clase *RST_PURPOSE*, subclase de la categoría *RHETORICAL_RELATION*.

En la categoría *RHETORICAL_RELATION* están definidas un conjunto de relaciones descritas en la Teoría de Estructura Retórica de Mann y Thompson (1987a, 1987b, 1988). En esta teoría, las relaciones mantenidas entre los distintos segmentos del texto son definidas en términos de «núcleos» y «satélites». Los elementos nucleares son aquellos esenciales para la realización de una relación, mientras que los elementos satélites sirven para apuntar los aspectos auxiliares. La clase *RST_PURPOSE*, por ejemplo, se aplica en los casos en que la actividad especificada en «*domain*» es ejecutada con el propósito de realizar la actividad presentado en «*range*». Esta clase pertenece a la categoría de *ASYMMETRIC_RHETORICAL_RELATION*, en la que uno de los participantes es el elemento nuclear («*domain*») y el otro el satélite («*range*»).

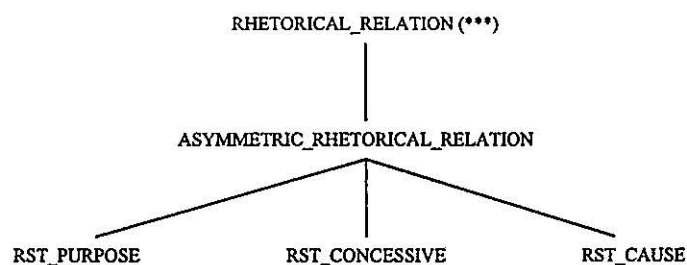


Figura 4.6.— Organización en el upper-model de las clases derivadas de la clase *RHETORICAL_RELATION*.

A esta categoría pertenecen otras clases que describen este tipo de relación entre dos entidades conceptuales. La clase *RST_CONCESSIVE* expresa la relación entre dos situaciones aparentemente incompatibles que deben ser aceptadas como compatibles (e.g., "aunque estaba muy cansado, se fue de copas"). *RST_CAUSE* se utiliza para describir la relación mantenida entre una acción («*domain*») y la situación que ha causado dicha acción («*range*») (e.g., "se fue a casa porque estaba muy cansado").

Una última categoría de *TWO_PLACE_RELATION* considerada en nuestro upper-model es *INTENSIVE*. En ella están integradas las relaciones de dos entidades que son idénticas

(*IDENTITY*), donde una simboliza a la otra (*SYMBOLIZATION*) y las relaciones de atribución (*ASCRPTION*).

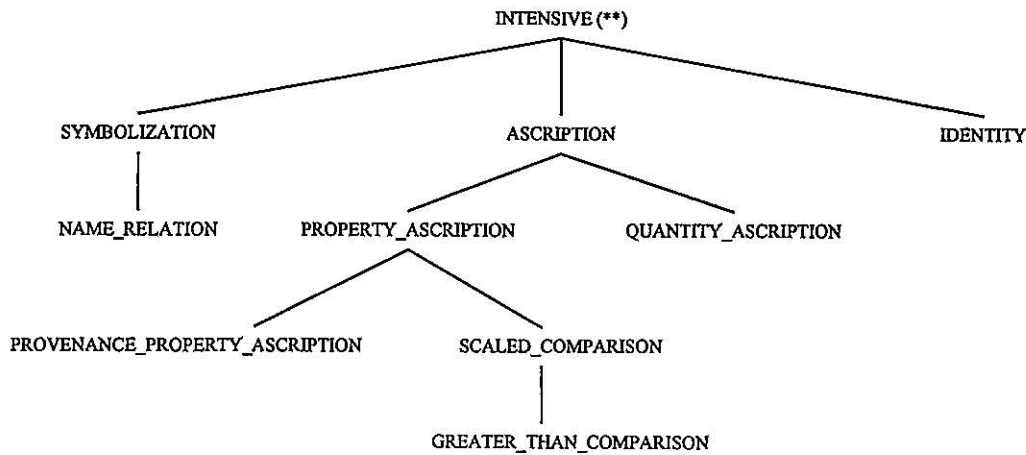


Figura 4.7.—Organización en el upper-model de la subjerarquía *INTENSIVE*.

Una especialización de la clase *SYMBOLIZATION* es la relación entre un objeto y su nombre (*NAME_RELATION*). De las relaciones de atribución se distinguen las de cantidad (*QUANTITY_ASCRPTION*) y las de propiedad (*PROPERTY_ASCRPTION*). De esta última, se derivan la clase *PROVENANCE_PROPERTY_ASCRPTION*, en la que la propiedad atributo del objeto es un gentilicio y la clase *SCALED_COMPARISON* que compara un objeto con respecto a una propiedad especificada en un grado menor o mayor (*GREATER_THAN_COMPARISON*).

4.2.1.1.2. La subjerarquía *OBJECT*

A esta categoría pertenecen todas las entidades conceptuales que hacen referencia a cosas, sean éstas abstractas o concretas, mentales o físicas. Una primera clasificación de los «objetos» se establece según su "consciencia". Por un lado, la clase *CONSCIOUS_BEING* describe entidades activas con capacidad de producir información y a las que se les atribuyen la propiedad de "consciencia". Generalmente, se trata de una persona (*PERSON*), bien sea hombre (*MALE*) o mujer (*FEMALE*). Por otro lado, están las cosas sin "consciencia" (*NON_CONSCIOUS_THING*). Dos clases derivadas están representadas en el upper-model: *ABSTRACTION*, «objetos» que existen en un sentido metafórico o en un plano cualitativo más que físico (e.g., "verdad"), y *SPATIAL_TEMPORAL*, categoría general que engloba todos los «objetos» espaciales (*SPATIAL*) y temporales (*TEMPORAL*).

Los «objetos» espaciales pueden ser considerados como un punto (*SPACE_POINT*, *ZERO_D_LOCATION*), una línea o un plano (*SPACE_INTERVAL*, *ONE_OR_TWO_D_LOCATION*). En cuanto a los «objetos» temporales, están presentes en nuestro *upper-model* dos clases de escalas temporales (*TIME_INTERVAL*): *ONE_OR_TWO_D_TIME* (e.g., "día") y *THREE_D_TIME* (e.g., "año").

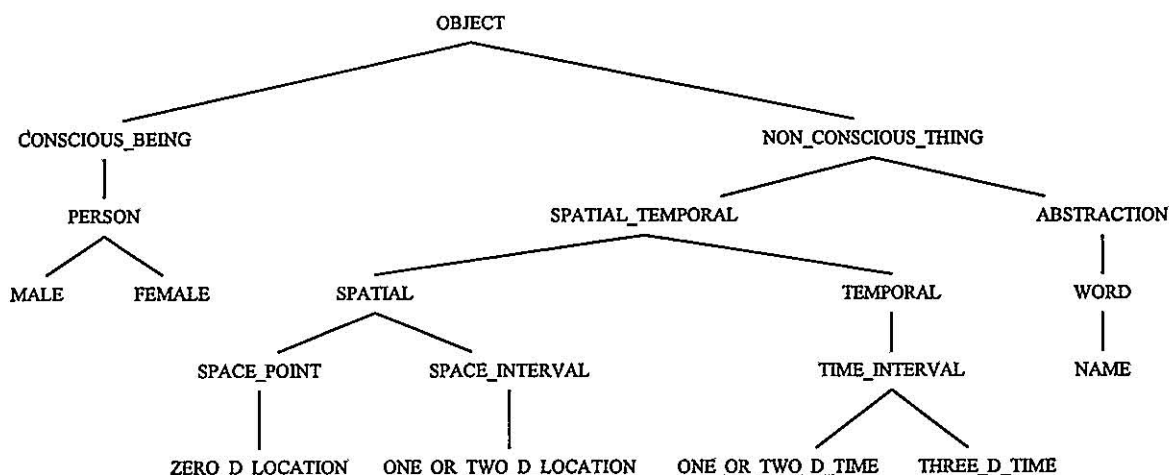


Figura 4.8.—Organización en el *upper-model* de la subjerarquía *OBJECT*.

Se han definido como atributos comunes a todas las clases de esta subjerarquía su identificabilidad («*identifiability*»), su número («*number*»), algunas relaciones específicas entre un elemento y el «objeto» en cuestión («*property_ascription*», «*generalized_possession*») y relaciones no específicas («*relations*», «*processes*»). Asimismo, la clase *PERSON* describe «objetos» con una propiedad particular, el género («*gender*»), que constituye un atributo adicional para la definición de esta clase en particular.

4.2.1.1.3. La subjerarquía *QUALITY*

Son entidades conceptuales que describen propiedades de «objetos» (*MATERIAL_WORLD_QUALITY*) y de «procesos» (*MODAL_QUALITY*).

El tamaño es una propiedad de los «objetos» que se caracteriza por ser una «cualidad» gradual, esto es, admite diferentes niveles o grados. Considerando este aspecto, son definidas las clases *SCALABLE_QUALITY* y *NONSCALABLE_QUALITY*. Una «cualidad» gradual como el tamaño (*SIZE*) es un caso particular de la categoría *SENSE_AND_MEASURE_QUALITY*. A ella pertenecen todas las propiedades que pueden ser percibidas o medidas por una entidad de la

clase *CONSCIOUS_BEING*. De la clase *SCALABLE_QUALITY* se deriva otra clase: *EVALUATIVE_QUALITY*. Ésta es un categoría que describe propiedades determinadas por un «ser consciente» según algún sistema de valores (e.g., "fácil", "bonito", "honesto").

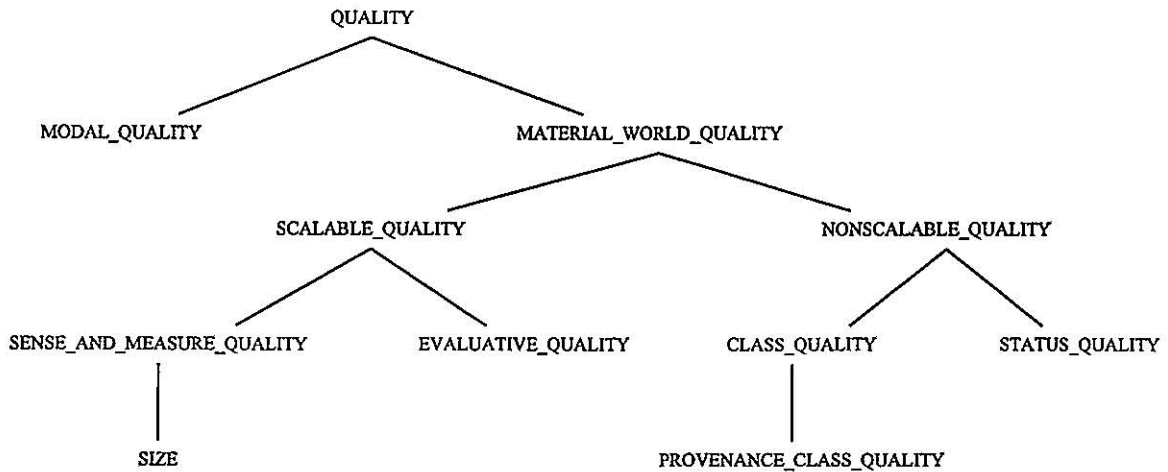


Figura 4.9.— Organización en el upper-model de la subjerarquía "QUALITY".

La categoría *NONSCALABLE_QUALITY* tiene dos especializaciones: la clase *STATUS_QUALITY* que describe propiedades de un objeto independiente del observador (e.g., "vivo", "muerto") y la clase *CLASS_QUALITY* que contiene distintas taxonomías como el origen de las personas (*PROVENANCE_CLASS_QUALITY*).

Con esto concluimos la descripción del componente *upper-model* del sistema. Todas las entidades conceptuales aquí descritas pertenecen a la versión desarrollada para PENMAN, en la que, además, están definidas otras entidades no consideradas en nuestro sistema (véase Bateman, Kasper, Moore y Whitney, 1990).

4.2.1.2. Domain-Model

Si las entidades conceptuales que integra el *upper-model* son independientes de dominio, en el *domain-model* están definidas todas las clases de objetos, acciones, relaciones, etc., presentes en el dominio específico de aplicación del generador. Estas entidades deben estar organizadas en estructuras jerárquicas donde los elementos raíces estén obligatoriamente subordinados a, al menos, una clase del *upper-model*.

Este componente del sistema contiene un importante conjunto de información. Junto al *upper-model*, constituye el medio de comunicación entre los datos de entrada y el generador. Todos los tipos de elementos conceptuales especificados en las entradas del sistema deben estar definidos en el *upper-model* o en el *domain-model*.

Por tanto, se ha definido un conjunto de entidades para la representación del contenido conceptual de los "mensajes preverbales" a partir de los cuales el generador construye el formato apropiado para su expresión lingüística. A modo ilustrativo, haremos referencia a algunas de estas entidades. Por ejemplo, *Landolfo_O* es el elemento conceptual que describe el personaje principal de la historia que conforma nuestro dominio particular: una persona de género masculino con nombre "Landolfo". Su vínculo con el *upper-model* es directo, esto es, es un descendiente inmediato de una clase del *upper-model*, la clase *MALE*. A continuación presentamos de forma gráfica la relación de algunas de las entidades de nuestro *domain-model* con el *upper-model*, a modo ilustrativo (para consultar otras entidades definidas en el *domain-model*, ver Anexo).

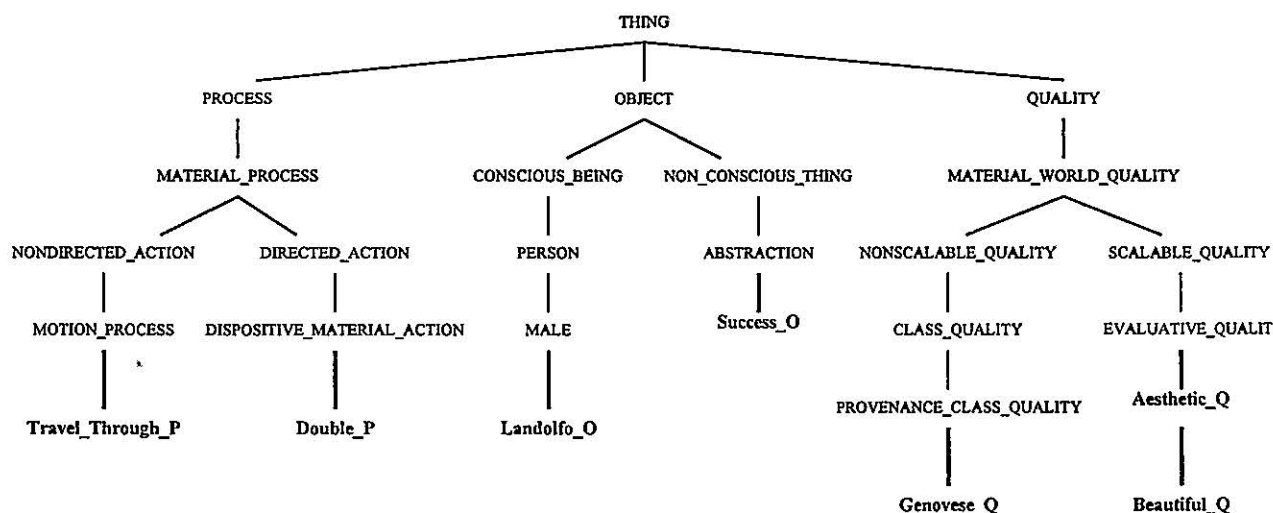


Figura 4.10.— Ejemplo de relación entre entidades del *domain-model* (con formato negrita) con el *upper-model*.

Todos los elementos definidos en el *domain-model*, así como algunos de los elementos del *upper-model*, deben estar asociados con al menos un ítem léxico contenido en el *lexicon* del sistema, componente que pasamos a describir en el siguiente epígrafe.

4.2.1.3. *Lexicon*

Los elementos terminales de una expresión lingüística son las palabras. Un sistema de generación de oraciones debe, por tanto, contar con información sobre los elementos léxicos del lenguaje natural destino del generador. Esto es, el «lexicon». Este componente de los generadores contiene las definiciones de los elementos léxicos (su categoría gramatical, sus componentes morfológicos, su ortografía, etc.). No es nuestro objetivo listar todos los elementos léxicos que integran el «lexicon» de nuestro sistema, sino describir la organización de estos elementos y su estructura de representación.

Todos los ítems léxicos "conocidos" por GEDEON pertenecen a una categoría de la jerarquía *Lexicon* y a una/s categoría/s de la jerarquía *Affix*. Cada una de ellas define un subconjunto específico de propiedades de los elementos léxicos. Por ejemplo, la clase *HERMOSO_A* es descendiente de las clases *Adjective*, *S_Number_Affix* y *OA_Gender_Affix* indicando su categoría gramatical y de flexión.

4.2.1.3.1. La jerarquía *Lexicon*

La clase base de esta jerarquía es *Lexicon* y tiene dos descendientes directos: *Phrase* y *Word*. La categoría *Phrase* contiene expresiones perifrásticas como "con el fin de" o "por esta razón" y *Word* es la clase de la que se derivan todas las palabras de la lengua castellana implementadas en nuestro generador.

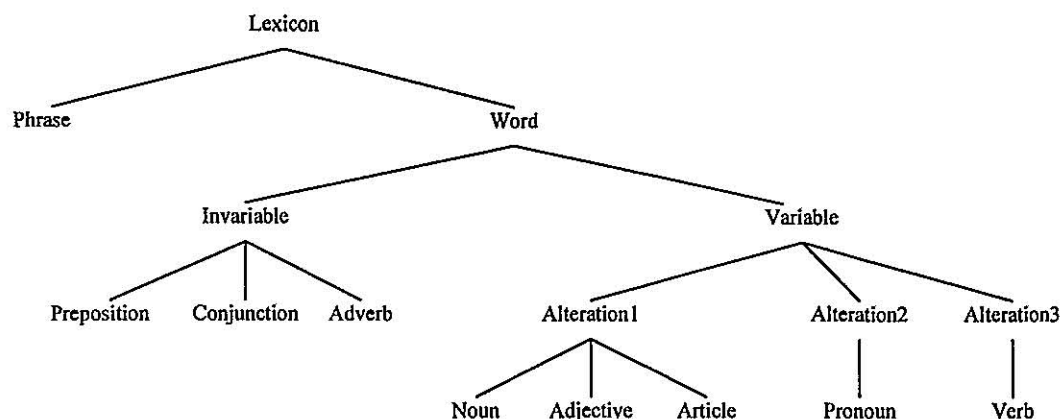


Figura 4.11.—Estructura de la jerarquía de las categorías gramaticales integradas en el *Lexicon*.

Las categorías gramaticales consideradas para agrupar los distintos elementos léxicos son: *Preposition*, *Conjunction*, *Adverb*, *Noun*, *Adjective*, *Article*, *Pronoun* y *Verb*. Estas categorías están definidas por propiedades particulares, en algunos casos compartidas por varias de ellas. Una primera distinción observada en la subjerarquía *Word* es su característica morfológica. Por un lado, preposiciones, conjunciones y adverbios son clases de palabras que no requieren de la diferenciación entre los morfemas raíz y afijo (*Invariable*). Por otro lado, la clase *Variable* se define por la presencia de dos atributos adicionales: «*root*» y «*affix*». De esta clase son derivadas tres subclases que se caracterizan por los tipos de accidentes gramaticales admitidos. Así, la clase *Alteration1* es definida por el género («*gender*») y el número («*number*»). A esta categoría pertenecen los nombre, adjetivos y artículos. La clase *Alteration2* contiene, además de género y número, la persona («*person*»), siendo los pronombres la clase de palabras que requieren de estas propiedades en su definición. En una tercera clase derivada de la clase *Variable*, *Alteration3*, se define las alteraciones admitidas en la conjugación verbal incluyendo la voz («*voice*»), el modo («*mode*»), el tiempo («*tense*»), el número («*number*») y la persona («*person*»).

Asimismo, se han incorporado algunas clases especializadas para las categorías *Noun*, *Pronoun* y *Verb*. Lejos de ser exhaustiva, en esta clasificación están contempladas únicamente aquellas subcategorías de nombre, pronombre y verbos requeridos para la expresión lingüística de los contenidos conceptuales presentes en nuestro dominio.

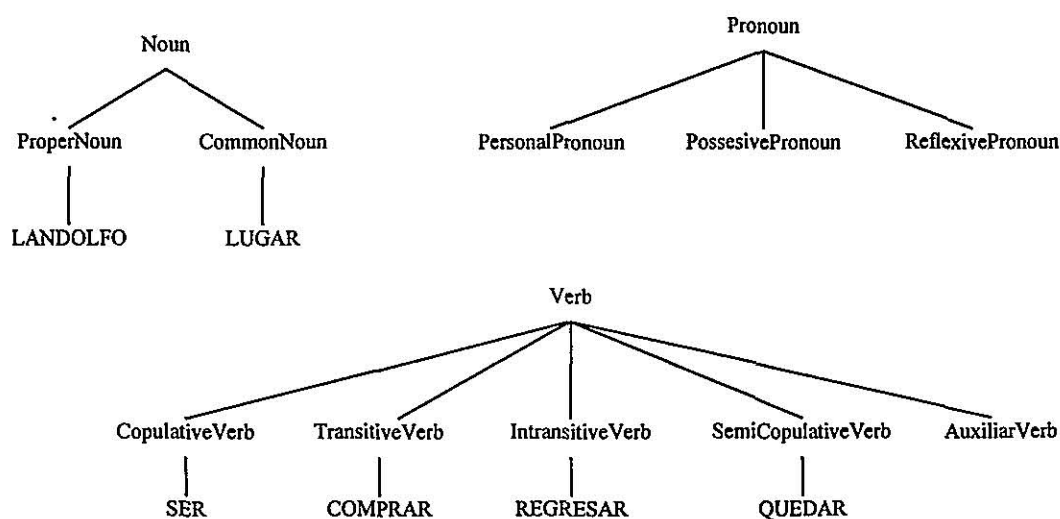


Figura 4.12.—Ejemplos de elementos léxicos considerados en las categorías *Noun*, *Pronoun* y *Verb*.

4.2.1.3.2. La jerarquía *Affix*

En esta jerarquía están definidos las distintas clases de flexión requeridas por el sistema. Dos clases derivadas de *Affix* engloban las alteraciones que afectan a las categorías de elementos léxicos que pertenecen a la clase *Alteration1* (*Alteration1_Affix*), por un lado, y a la clase *Alteration3* (*Alteration3_Affix*), por otro.

La clase *Alteration1_Affix* se deriva en dos subclases: *Number_Affix* y *Gender_Affix*, que corresponde a las flexiones de los accidentes gramaticales número y género, respectivamente. En cuanto al número, tres casos particulares de formación del plural han sido considerados para este sistema: añadiendo una "s" (*S_Number_Affix*), añadiendo "es" (*ES_Number_Affix*) y añadiendo "es" después de sustituir "z" por "c" (*CES_Number_Affix*). Ejemplos de cada una de estas categorías son, respectivamente: "costa"/"costas", "lugar"/"lugares", "capaz"/"capaces".

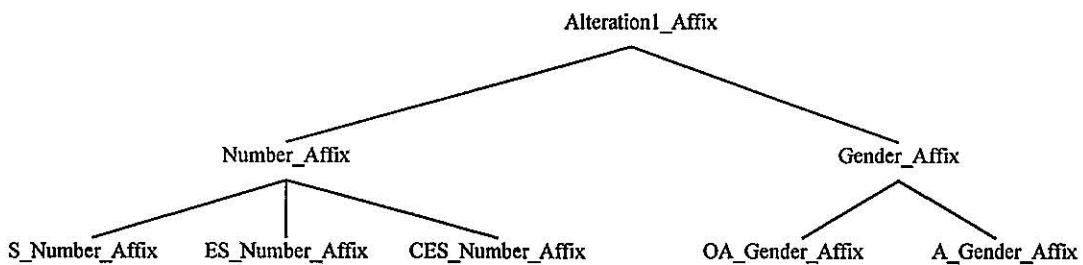


Figura 4.13.—Organización de la subjerarquía *Alteration1_Affix*.

Respecto a la clase *Gender_Affix*, se han definido dos subclases que corresponden a la formación del femenino sustituyendo "o" por "a" (*OA_Gender_Affix*) o añadiendo una "a" (*A_Gender_Affix*).

En la categoría *Alteration3_Affix* están representadas distintas formas que pueden tomar los verbos, en virtud de los accidentes gramaticales. Un primer nivel de clasificación distingue las tres conjugaciones regulares básicas: *AR_Regular_Affix*, *ER_Regular_Affix*, *IR_Regular_Affix*, en las que están definidas las flexiones correspondientes a la primera, la segunda y la tercera conjugación de los verbos regulares.

Algunas de las formas verbales irregulares han sido incorporadas en el sistema. En estos casos, la clasificación adoptada se corresponde a la propuesta por Miranda Podadera en su libro *Análisis Gramatical* (1984). Es importante observar que las distintas clases de conjugación

irregular consideradas en el sistema se derivan, en todos los casos, de una de las formas regulares. Esto es debido a que las irregularidades no suelen afectar a todas sus conjugaciones. Así, las dos subclases de la primera conjugación regular *AR_Irregular_Spelling_C_Affix* y *AR_Irregular_Spelling_G_Affix* están representadas en las alteraciones obligadas por las reglas ortográficas que afectan, por ejemplo, para el pretérito indefinido de indicativo, únicamente a la conjugación correspondiente a la primera persona del singular (e.g., "atacar"-*"ataqué"* y "llegar"-*"llegué"*). Por tanto, en estas subcategorías están definidas únicamente las formas alteradas de conjugación. Para los tiempos y personas que se ajustan a la conjugación regular, el sistema aplica las formas definidas en la categoría de la que se derivan.

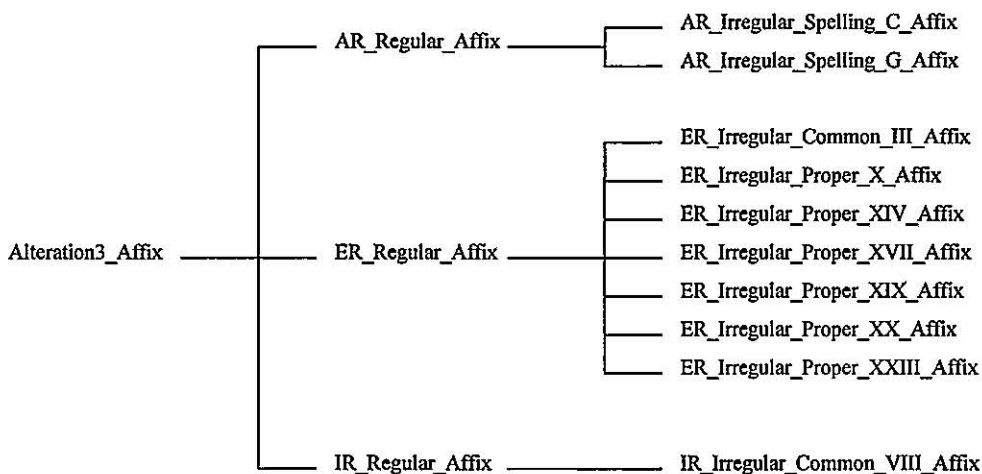


Figura 4.14.— Organización de la subjerarquía *Alteration3_Affix*.

Para verbos de la segunda conjugación se han definido las siguientes clases de conjugación irregular:

⊠ *ER_Irregular_Common_III_Affix*. A los verbos que pertenecen a este grupo (e.g., "conocer") se añade una "z" antes de la "c" radical en la conjugación de la primera persona singular del presente de indicativo (e.g., "conozco").

⊠ *ER_Irregular_Proper_X_Affix*. Un caso específico de esta clase es el verbo "hacer". Las formas alteradas de conjugación para el modo indicativo son: primera persona singular del presente ("hago"), pretérito indefinido ("híce", "hiciste", "hizo", "hicimos", "hicisteis", "hicieron") y futuro imperfecto ("haré", "harás", "hará", "haremos", "haréis", "harán").

✧ *ER_Irregular_Proper_XIV_Affix*. A este grupo pertenece, por ejemplo, el verbo "poder" con las siguientes alteraciones: gerundio ("pudiendo"), presente de indicativo ("puedo", "puedes", "puede", "pueden"), pretérito indefinido ("pude", "pudiste", "pudo", "pudimos", "pudisteis", "pudieron") y futuro imperfecto de indicativo ("podré", "podrás", "podrá", "podremos", "podréis", "podrán").

✧ *ER_Irregular_Proper_XVII_Affix*. El verbo que nos va servir para ilustrar esta clase es el verbo "querer". Las formas alteradas consideradas son el presente de indicativo ("quiero", "quieres", "quiere", "quieren"), el pretérito indefinido ("quise", "quisiste", "quiso", "quisimos", "quisisteis", "quisieron") y el futuro imperfecto de indicativo ("querré", "querrás", "querrá", "querremos", "querréis", "querrán").

✧ *ER_Irregular_Proper_XIX_Affix*. En ella se define la conjugación del verbo "ser".

✧ *ER_Irregular_Proper_XX_Affix*. Alteraciones del verbo "tener" ("obtener" es otro ejemplo) en el presente de indicativo ("tengo", "tienes", "tiene", "tienen"), el pretérito indefinido ("tuve", "tuviste", "tuvo", "tuvimos", "tuvisteis", "tuvieron") y futuro imperfecto de indicativo ("tendré", "tendrás", "tendrá", "tendremos", "tendréis", "tendrán").

✧ *ER_Irregular_Proper_XXIII_Affix*. Describe las irregularidades de verbos como "ver" con alteraciones en el presente ("veo") y pretérito imperfecto de indicativo ("veía", "veías", "veía", "veíamos", "veíais", "veían").

Finalmente, a la clase *IR_Irregular_Common_VIII_Affix* pertenecen verbos que diptongan en "ie" la "e" de la penúltima sílaba ("convertir"). Son irregulares en las formas de gerundio ("convirtiendo"), presente de indicativo ("convierto", "conviertes", "convierte", "convierten") y pretérito indefinido ("convirtió", "convirtieron").

4.2.1.4. Grammar

Un generador de oraciones en lenguaje natural debe "conocer", además, las reglas que organizan las palabras en una secuencia ordenada, conformando distintas estructuras sintácticas que sirven para la expresión lingüística de las distintas piezas conceptuales de la representación del contenido del mensaje y las relaciones establecidas entre ellas. El conocimiento de estas reglas constituye el componente gramatical del sistema: el *Grammar*.

En este módulo están definidos un conjunto de *especialistas sintácticos* que tienen la función de construir el plan adecuado para la realización lingüística de la representación conceptual recibida. Este plan consiste, básicamente, en una secuencia ordenada de metas que el sistema debe alcanzar y constituye, generalmente, el marco sintáctico en el que se ensamblan los elementos léxicos.

La complejidad de los *especialistas* varía según la cantidad de opciones lingüísticas de que dispone para la construcción de un marco sintáctico. Esto es, el número de planes alternativos que el *especialista* puede generar. Sin embargo, no hay una correspondencia exacta entre los *especialistas* y las categorías de estructuras sintácticas descritas en la gramática. De hecho, la definición de los *especialistas* responde, fundamentalmente, a criterios funcionales: cualquier secuencia de operaciones que crea un marco apropiado para la expresión lingüística de una determinada entidad conceptual es definido como un *especialista*.

A modo ilustrativo, vamos a presentar algunos de los *especialistas sintácticos* implementados en el sistema GEDEON. Por ejemplo, la función *Say_Relation_Sent_01* genera la siguiente secuencia de metas: {*Say_Sentence* + *Say_Conjunction* + *Say_Sentence*}. Este *especialista* es el responsable de generar oraciones coordinadas como "vendió su casa y se fue de la ciudad".

El *especialista* que crea el marco sintáctico para la generación de la oración "vendió su casa" es *Say_Propositional_Sent_04*. A diferencia de *Say_Relation_Sent_01*, este *especialista*, dispone de diversas opciones lingüísticas que deben ser consideradas para la elaboración del plan. Algunos de estos planes están definidos por estas secuencias de metas:

{[*Say_Subject*] + *Say_Verb* + [*Say_Complement*]}³
{[*Say_Subject*] + *Say_Verb* + *Say_Object* + [*Say_Complement*]}
{*Say_Complement* + *Say_Mark* + *Say_Subject* + *Say_Verb* + *Say_Complement*}
{[*Say_Subject*] + *Say_Verb* + *Say_Attribute*}

En el caso de la oración "vendió su casa", *Say_Propositional_Sent_04* genera el plan {*Say_Verb* + *Say_Object*}. El complemento directo de esta oración es generado por la

³ Los elementos entre corchetes son opcionales.

intervención de los *especialistas Say_Object_02* y *Say_Nominal_Group_02* que son definidos, respectivamente, de las siguientes formas:

$\{Say_Nominal_Group\}$

$\{[Say_Determiner] + [Say_Premodifier] + Say_Noun + [Say_Posmodifier]\}$

El plan específico que faculta la generación del sintagma nominal "su casa" consta de dos metas: $\{Say_Determiner + Say_Noun\}$. La ausencia de algunas de las metas opcionales se debe, en este caso, a los valores NULL de los atributos *property_ascription*, *relations* y *processes*. Esto es, las entidades conceptuales y sus relaciones son algunos de los criterios considerados por los *especialistas* a la hora de construir los planes. Sin duda, no son los únicos, como veremos en la descripción del módulo *Input* del sistema.

4.2.1.5. Goals

Como hemos comentado al principio de este capítulo, el sistema GEDEON es un generador táctico "dirigido por metas". Esto exige la definición de un conjunto de «metas» que constituye el componente *goals* de nuestro sistema. Cada «meta» cuenta con una función de evaluación asociada y la información requerida para la realización de dicha función. La estructura de representación apropiada es, por tanto, la clase (propiedad de encapsulamiento). Guiándonos por el tipo de información incorporado a las clases de «metas», el módulo *Goals* describe la estructura jerárquica presentada en la *Figura 4.15*.

Una primera diferenciación entre las «metas» observada en la jerarquía se refiere al tipo de información requerida para su realización. Las clases derivadas de la categoría *Syntax_Goal* se caracterizan por la presencia de una representación conceptual que restringe, si no determina, los *especialistas sintácticos* aplicables en cada caso. Sobre estas representaciones conceptuales operan los *especialistas* solicitados para la elaboración de nuevas secuencias de «metas».

Por su parte, la clase *Lexical_Goal* tiene como atributo asociado el elemento léxico seleccionado para la expresión lingüística de una entidad conceptual determinada. Este atributo es heredado por todas las subclases derivadas de ella así como una función *Choose_Plan* definida para el ámbito de *Lexical_Goal*. Esta función recupera la representación ortográfica del ítem léxico especificado y genera un único plan con la «meta» *Say_Word*. Para casos particulares de realización de «metas» de la categoría *Lexical_Goal*, la función *Choose_Plan* es

redefinida para el ámbito de la subclase en la que se produce la excepción. Por lo general, esta función genera planes que consisten en secuencias de «metas» *Say_Word* (e.g. "con el fin de").

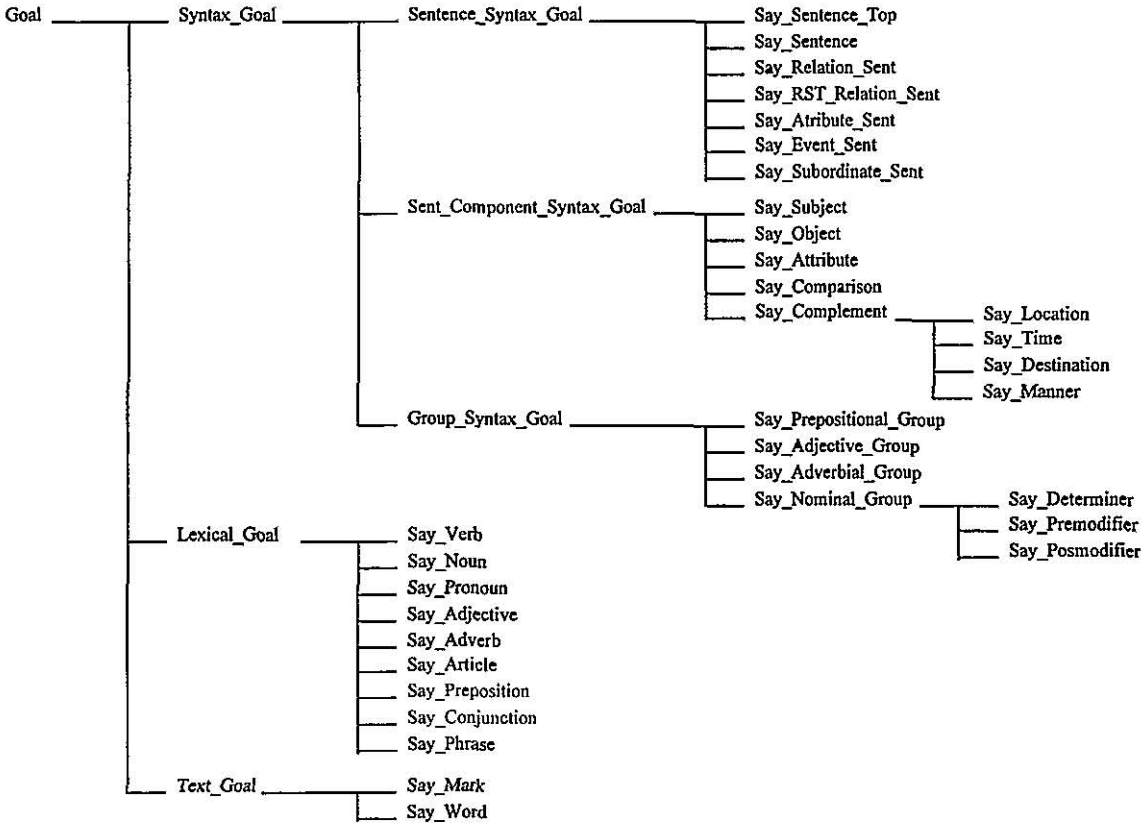


Figura 4.15.—Representación gráfica de la estructura de la jerarquía Goal.

Finalmente, las clases *Say_Mark* y *Say_Word* contienen, respectivamente, la representación ortográfica de un signo de puntuación y de una palabra. La función definida para la realización de estas «metas» consiste en solicitar la presentación de la información ortográfica como salida del sistema.

Hasta aquí, hemos descrito los aspectos básicos de las «metas» definidas en nuestro sistema. Vamos a abordar ahora las particularidades expuestas por algunas de ellas. Empezaremos con las «metas sintácticas».

Las subjerarquías establecidas para la categoría *Syntax_Goal* responden a dos criterios de clasificación: el ámbito sintáctico en consideración y la clase de información adicional requerida. La clase *Sentence_Syntax_Goal* agrupa las «metas» del sistema que demanda la construcción de estructuras sintácticas de niveles superiores como oraciones y proposiciones.

De las clases definidas dentro de esta categoría, la clase *Say_Subordinate_Sent* es la única que contiene un atributo adicional, el «link»: ítem léxico que sirve como nexo de la proposición subordinada con la proposición principal.

La clase *Sent_Component_Syntax_Goal* es definida para integrar las «metas» que el sistema utiliza para asignar los distintos papeles funcionales que las entidades conceptuales realizan dentro del marco sintáctico de una proposición. Todas ellas contienen los elementos léxicos correspondientes a los núcleos del sujeto y del predicado como sus atributos adicionales.

Las «metas» derivadas de la clase *Group_Syntax_Goal* corresponden a categorías de constituyentes sintagmáticos. Para cada una de ellas se ha definido un atributo adicional que contiene la información léxica relevante para su realización. En el caso de *Say_Nominal_Group*, la información adicional está representada por un ítem léxico específico de la clase *Noun*. De esta clase se derivan las subclases *Say_Determiner*, *Say_Premodifier* y *Say_Posmodifier* que comparten una misma información léxica heredada de la clase superior. Hemos hecho referencia explícita sobre estas clases para ilustrar el método empleado por el sistema para llevar a cabo la concordancia de género y número entre el núcleo y los modificadores de un sintagma nominal.

Por último, volvemos a los caso particulares de la subjerarquía *Lexical_Goal* que, como ya hemos mencionado, se caracterizan por requerir más de una «meta» *Say_Word* para su realización. La clase *Say_Phrase* es el ejemplo más evidente. El elemento léxico que define esta clase pertenece a la categoría *Phrase* del componente *lexicon*. Esto es, la categoría de las expresiones perifrásticas que por definición están compuestas por más de una palabra (e.g., "con el fin de").

La clase *Say_Verb* constituye otro caso de excepción para la categoría *Lexical_Goal*. En este caso, la función de realización asociada debe considerar algunos aspectos adicionales en la elaboración del plan, como por ejemplo, el índice reflexivo. Esto es, si el uso del verbo es reflexivo, sea propio (i.e., verbos que nunca pueden ser más que reflexivos) o impropio (e.g., "suicidarse" e "irse", respectivamente), la función genera un plan con dos «metas» *Say_Word* que corresponden al verbo y al pronombre reflexivo (e.g., "se fue").

4.2.1.6. Input

El proceso de generación de oraciones ejecuta sus operaciones en función de las especificaciones contenidas en una representación de entrada (i.e., mensaje preverbal). En un sistema de generación textual, esta representación es, a su vez, la salida producida por el generador estratégico (i.e., conceptualizador). En ella están definidos todos los elementos conceptuales del contenido del mensaje acompañados por criterios interpersonales y textuales imprescindibles para su realización lingüística. Juntos constituyen la clase *Topic* del sistema.

Todos los «temas» considerados por el generador están definidos por una entidad conceptual del *upper-model* o del *domain-model*. Se ha observado que elementos de cada una de las subjerarquías conceptuales fundamentales (*Process*, *Object*, *Quality*) tienen asociados criterios interpersonales y textuales particulares. Con esto, se han descrito tres subclases de *Topic* en el sistema. La clase *Process_Topic* cuenta con los siguientes criterios adicionales: «*speechact*» (tipo de acto de habla -aseveración, orden, pregunta, etc.-); «*polarity*» (indicador del signo de la expresión lingüística -afirmación o negación-); «*thema*» (marcador de la entidad conceptual dominante); y «*tense*» (contenedor de un indicador del tiempo verbal).

La clase *Object_Topic*, por su parte, tiene un único atributo adicional, «*expression_type*», en el que se especifica el modo de expresión lingüística del elemento conceptual. Este es el criterio que determina el uso de sujeto elíptico o su pronominalización en orden a la cohesión textual. Para una tercera clase, *Quality_Topic*, no se ha descrito ningún atributo adicional. Se caracteriza únicamente por contener elementos conceptuales de la categoría *Quality*.

Veamos un ejemplo de representación inicial del sistema, considerando la estructura conceptual presentada en la *Figura 4.1.*:

```

(TopicA / Process_Topic
: thing (ThingA / Reside_P
: actor (TopicB / Object_Topic
: thing (ThingB / Merchant_O
: identifiability NO
: number SINGULAR
: gender MASCULINE
: property-ascription (Topic4 / Process_Topic
: thing (Thing4 / PROPERTY_ASRIPTION
: domain TopicB
: range (TopicG / Quality_Topic
: thing (ThingG / Rich_Q).)
: relations (Topic5 / Process_Topic
: thing (Thing5 / NAME_RELATION
: domain TopicB
: range (TopicH / Object_Topic
: thing (ThingH / LANDOLFO_O).)
: expression-type YES)
: location (TopicC / Object_Topic
: thing (ThingC / ZERO_D_LOCATION
: identifiability NO
: number SINGULAR
: property-ascription (Topic1 / Process_Topic
: thing (Thing1 / PROPERTY_ASRIPTION
: domain TopicC
: range (TopicD / Quality_Topic
: thing (TopicD / Beautiful_Q).)
: relations (Topic3 / Process_Topic
: thing (Thing3 / PART_OF
: domain TopicC
: range (TopicE / Object_Topic
: thing (ThingE / Coast_O
: identifiability YES
: number PLURAL
: relations (Topic2 / Process_Topic
: thing (Thing2 / PART_OF
: domain TopicE
: range (TopicF / Object_Topic
: thing (ThingF / Italia_O).)
: event-tense PAST1)
: speechact ASSERTION
: polarity POSITIVE
: theme LOCATION)

```

La notación básica utilizada para la descripción de la representación de entrada tiene el siguiente formato:

(identificador de la instancia / Nombre de la clase
: atributo Valor*)

donde, * denota cero o más ocurrencia del término ": atributo Valor".

Así, observamos en el ejemplo que **TopicA** es el identificador de una instancia⁴ de la clase *Process_Topic*. Es definido por los atributos «thing», «speechact», «polarity» y «thema» con los valores **ThingA**, **ASSERTION**, **POSITIVE** y **LOCATION**, respectivamente. **ThingA** es el identificador de la entidad conceptual *Reside_P* definido para esta representación de entrada

⁴ Instancia es, en la programación orientada a objetos, un objeto particular de una clase con valores específicos para los atributos que la definen.

particular que cuenta con los «temas» **TopicB** y **TopicC** como valores asignados a los atributos «actor» y «location». Cada uno de estos «temas» contiene el elemento conceptual correspondiente que, asimismo, puede estar definido por instancias de subclases de *Topic*.

4.2.1.7. *Output*

En el lenguaje hablado, este módulo constituye un campo específico de investigación tanto en la psicolingüística como en la inteligencia artificial. La complejidad de los sistemas de generación del habla es comparable a la de los generadores de lenguaje natural que optan por el texto escrito como su formato de presentación de salida.

Una vez que el sistema de realización lingüística recupera la representación ortográfica de las palabras, operación imprescindible para alcanzar cualquier «meta» de la categoría *Lexical_Goal*, la tarea del módulo *Output* se muestra especialmente sencilla: presentar la cadena de caracteres contenida en dicha representación en un dispositivo de salida como es el monitor. Esto es, una simple línea de código C: `printf ("%s", cadena)`. Sin embargo, la función de presentación de salida resulta algo más compleja.

Consideremos por un instante el uso de las contracciones en el castellano. Este fenómeno de unir dos palabras en una, con pérdida de alguna vocal ("al", "del") requiere del sistema un tratamiento particular. Probablemente, éste es el único aspecto del sistema al que nos hemos enfrentado sin contar con opciones previamente consideradas por otros autores. Teniendo presente el diseño de nuestro sistema, esta cuestión podría haber sido tratada como: a) un caso particular de realización de sintagma preposicional; b) una categoría adicional del *Lexicon*; o, c) una forma especial de emisión lingüística. Considerando que nos encontramos describiendo el módulo *Output* del sistema, es evidente que hemos optado por esta última alternativa.

El módulo *Output* es, por tanto, un procedimiento de presentación de salida que consiste fundamentalmente en escribir en consola una cadena de caracteres salvo casos excepcionales como el de las contracciones. La aplicación de operaciones específicas para estos casos está determinada por la palabra previamente emitida y la palabra por emitir. Así, siguiendo con el ejemplo de las contracciones, el procedimiento cuenta, para los casos en que el par de palabras a evaluar es [a, el] o [de, el], con una operación particular de modificación de la cadena de caracteres a emitir que consiste en eliminar la letra "e" y el espacio en blanco introducido por

defecto como separador entre las palabras. Esta operación es ejecutada por el procedimiento antes de proceder a la emisión de la nueva palabra.

4.2.1.8. *Main-Process*

Este módulo constituye el centro de control de ejecución del sistema GEDEON. Como ya hemos comentado, para un generador de oraciones "dirigido por metas", el orden de evaluación de las metas tiene efecto directo sobre el orden de presentación de las palabras. Es, por tanto, imprescindible que el sistema mantenga un control explícito sobre la secuencia de metas a evaluar. En esto, hemos adoptado la estrategia utilizada por el realizador lingüístico del sistema PAULINE (Hovy, 1988): implementar una pila (*stack*).

La pila es una estructura de datos LIFO (*last in, first out*). Esto es, una lista de unidades de información en la que el último elemento introducido es el primero a extraer. En el sistema GEDEON, estas unidades de información son las «metas». El estado inicial de la pila es vacío (no contiene ninguna «meta»). A medida que el sistema va generando planes, estas secuencias de «metas» son introducidas en la pila en el orden inverso establecido. De esta forma, el primer elemento de la pila, esto es, la próxima «meta» a evaluar por el sistema, corresponde siempre a la primera «meta» del plan construido para alcanzar la «meta» evaluada, exceptuando los casos de «metas» *Text_Goal* (*Say_Mark* y *Say_Word*). En estos casos, *main-process* no recibe del procedimiento de evaluación asociado nuevas «metas» para ser introducidas en la pila.

A medida que las sucesivas expansiones de «metas» alcanzan «metas» *Text_Goal*, se produce una situación particular en la que la extracción de un elemento de la pila no va acompañada de la introducción de nuevos elementos en ella. Esto es lo que permite que una vez emitidas todas las palabras y signos de puntuación seleccionados la pila esté nuevamente vacía.

Así, el componente *main-process* de nuestro sistema de generación de oraciones es definido como un proceso recursivo de evaluación de «metas». Este proceso comienza con una «meta» inicial y finaliza cuando todas las «metas» generadas sucesivamente han sido evaluadas. Gráficamente, en un diagrama de flujo, el *main-process* se describe como sigue:

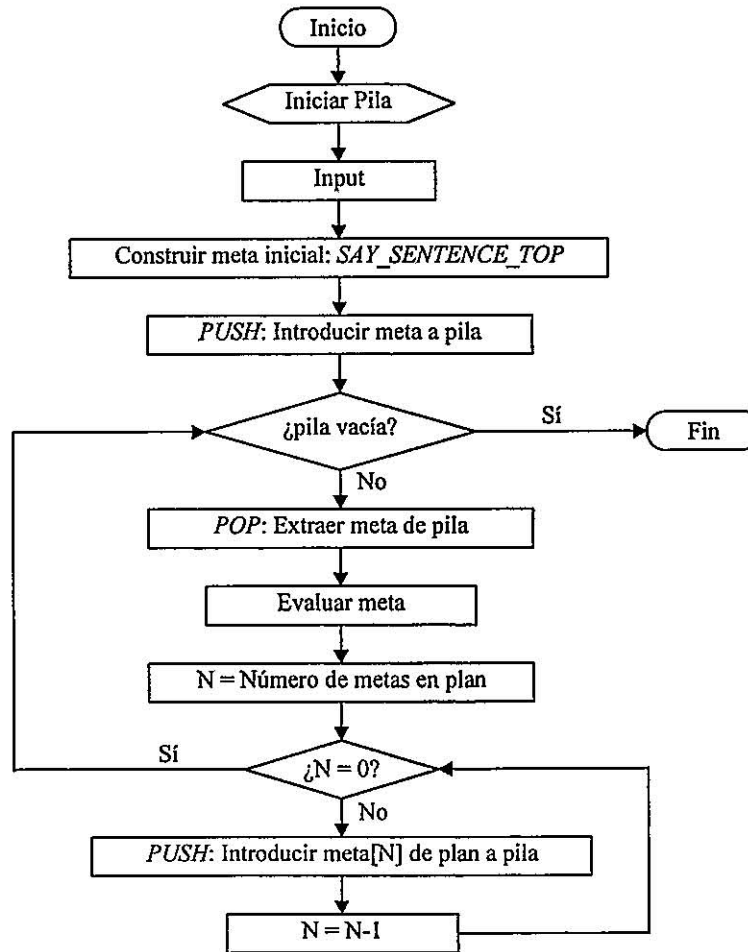


Figura 4.16.—Diagrama de flujo de las operaciones realizadas por el main-process.

4.2.2: Funcionamiento del sistema

En el sistema GEDEON, el proceso de generación de oraciones se describe como un proceso de evaluación de metas. Es, por tanto, imprescindible que todas las metas definidas en el sistema cuenten con su propia función de evaluación al que el módulo *main-process* invoca para obtener el plan apropiado para alcanzar dicha meta. En el sistema están definidos tres tipos de funciones de evaluación asociadas a las distintas clases de «metas»: funciones que elaboran planes específicos de realización, funciones que contemplan llamadas a *especialistas sintácticos* y funciones que invocan funciones de evaluación específicas asociadas a la categoría de la información conceptual para cada caso particular.

El proceso de generación comienza con la evaluación de la meta inicial, *Say_Sentence_Top*. La función asociada a esta «meta» elabora un único plan compuesto por las «metas» *Say_Sentence* y *Say_Mark*. Sin embargo, en la construcción de un plan para la «meta» *Say_Sentence*, la categoría de la entidad conceptual juega un papel fundamental. En estos casos, la función de evaluación asociada a la «meta» transfiere su cometido a funciones de evaluación asociadas a categorías conceptuales específicas en las que se definen procedimientos particulares de elaboración de planes. Así, por ejemplo, para la evaluación de la «meta» *Say_Sentence*, la función de evaluación asociada a la clase *INTENSIVE* ejecuta el *especialista sintáctico Say_Sentence_04* para la construcción del plan apropiado para alcanzar la «meta» mientras que el plan adecuado para la realización de un elemento conceptual de la clase *RHETORICAL_RELATION* es producto del *especialista Say_Sentence_03*.

Las nuevas «metas» que el sistema debe alcanzar y las submetas que de ellas se derivan son evaluadas sucesivamente en el orden establecido hasta que todas ellas hayan sido evaluadas, tal como hemos expuesto en el apartado 4.2.1.8. (páginas 107-108). En este proceso recursivo de evaluación, las funciones de evaluación y las operaciones que las definen intervienen a medida que las condiciones exigidas para su ejecución son alcanzadas. Muchas de estas condiciones hacen referencia a aspectos conceptuales de la representación de entrada. Por esta razón, tanto las funciones de evaluación como las operaciones que participan en la generación de oraciones, así como el momento del proceso en el que cada una realiza su intervención, están determinadas por características específicas presentes en el "mensaje preverbal".

Así, el proceso de generación, además de describirse como un proceso "dirigido por metas", es un proceso "guiado por contenido". Por tanto, resulta casi inevitable, reducir la descripción del funcionamiento del sistema a un caso particular para realizar una exposición detallada del proceso de generación. Este es el propósito de nuestro siguiente epígrafe.

4.2.3. Un ejemplo comentado de realización

Vamos a ilustrar el proceso de generación táctica con un ejemplo concreto que seguiremos paso a paso tomando como representación de entrada la presentada en el apartado 4.2.1.6. (página 105)⁵.

⁵ Consultar Anexo para seguir el ejemplo con la DEMO del sistema.

Como hemos expuesto anteriormente, la meta inicial del proceso es *Say_Sentence_Top*, a la que está asociada la representación conceptual de entrada. En este caso particular, la etiqueta que identifica dicha representación es **TopicA**. La evaluación de esta meta produce como resultado la siguiente secuencia de metas que son introducidas a la pila de metas pendientes en el orden establecido: *Say_Sentence* (con **TopicA** como elemento conceptual asociado) y *Say_Mark*.

A continuación, el *main-process* procede a extraer la primera de las metas de la pila e inicia la búsqueda de un plan para alcanzar dicha meta ejecutando la función de evaluación asociada, en este caso, a la meta *Say_Sentence*. Esta función deriva su labor a la función de evaluación asociada a la clase *PROCESS*, a cuya jerarquía pertenece la entidad conceptual de nuestra representación de entrada (**ThingA**). Esta función asigna el valor del atributo «event-tense» asociado a **ThingA** al atributo «tense» de **TopicA** y ejecuta la función *Say_Sentence_01*, *especialista sintáctico* encargado de construir el plan compuesto por una única meta, *Say_Event_Sent*. Este plan es recibido por el *main-process* que introduce la meta a la pila. Considerando que la pila cuenta con una meta pendiente, *Say_Mark*, su estado actual es el siguiente: *Say_Event_Sent* y *Say_Mark*.

El primer elemento de la pila es *Say_Event_Sent*, la próxima meta a evaluar por el sistema. Al igual que en el caso anterior, la función de evaluación asociada a la meta invoca a la función correspondiente asociada a la clase *PROCESS*. Esta función procede a la selección del ítem léxico apropiado para la expresión lingüística de la entidad conceptual **ThingA**, instancia particular de la clase *Reside_P*. En esta clase, como en todas las clases que componen el *domain-model* y algunas de las del *upper-model*, está definida una función de selección léxica que, en este caso, devuelve a la función de evaluación el ítem léxico *VIVIR*. Es importante tener presente la diferencia entre los elementos léxicos del componente *lexicon* y la representación ortográfica de estos elementos. En este punto del proceso, el sistema cuenta con una instancia de la clase *VIVIR* pero no "conoce" aún el formato ortográfico específico de la palabra que finalmente será emitida por el sistema.

Seleccionado el verbo principal de la oración, la función de evaluación ejecuta el *especialista sintáctico* *Say_Propositional_Sent_04*. La primera operación a realizar por esta función es la de asignar el modo, tiempo, persona y número, para lo cual utiliza distinta información contenida en **TopicA** («speechact», «tense») y en el elemento conceptual identificado como «actor» (**ThingB**). Así, en este ejemplo, los valores asignados a cada uno de

estos atributos del ítem léxico *VIVIR* son, respectivamente, *INDICATIVE*, *PAST1*, *THIRD* y *SINGULAR*.

La categoría del ítem léxico (*INTRANSITIVE_VERB*), los elementos circunstanciales asociados al elemento conceptual en cuestión y el valor del atributo «*expression-type*» de **TopicB** («tema» que realiza el papel de «*actor*») conducen la función a invocar un nuevo *especialista sintáctico* para la construcción del plan de realización.

El *especialista*, *Say_Propositional_Sent_08*, selecciona el ítem léxico correspondiente al elemento conceptual asignado como sujeto de la oración. En nuestro ejemplo dicho elemento conceptual es una instancia de la clase *Merchant_O* (**ThingB**) y el ítem léxico recuperado es *COMERCIANTE*. Comprobada, nuevamente, los elementos circunstanciales asociados a **ThingA**, el *especialista* devuelve a *main-process* la siguiente secuencia de metas:

Say_Complement
Say_Mark
Say_Verb
Say_Subject

Introducido el resultado de la evaluación de la meta *Say_Event_Sent* en la pila, el proceso continúa con la evaluación de la meta *Say_Complement* que tiene asociado como «tema» el elemento **TopicA**. La entidad conceptual asociada, **ThingA**, tiene como atributo «*location*» el «tema» etiquetado como **TopicC**. Siendo este atributo el único elemento circunstancial asociado a **ThingA**, la función de evaluación asociada a *Say_Complement* elabora un plan compuesto por la meta *Say_Location* con **TopicC** como «tema» asociado.

Veamos el nuevo estado de la pila:

Say_Location
Say_Mark
Say_Verb
Say_Subject
Say_Mark

Como podemos observar, el primer elemento de la pila es *Say_Location*, la meta que el sistema debe alcanzar en primer lugar. La evaluación de esta meta es realizada por la función asociada a la clase *OBJECT*. Esto es así porque el elemento conceptual **ThingC** asociado a nuestro «tema» es una instancia de la clase *ZERO_D_LOCATION* que pertenece a la subjerarquía *OBJECT* del *upper-model*. Esta función de evaluación asigna el ítem léxico *EN*

como enlace de la expresión lingüística derivada del elemento conceptual **ThingC** con el resto de la representación conceptual inicial (**ThingA**) y ejecuta el *especialista sintáctico* *Say_Location_01*. Como resultado de la evaluación, una nueva meta es introducida en la pila: *Say_Prepositional_Group*. Ésta, como último elemento introducido en la pila y, por tanto, primer elemento a extraer por el *main-process*, es nuestra siguiente meta a evaluar.

La elaboración del plan para alcanzar esta meta es, al igual que *Say_Sentence_Top* (comentado al principio de este epígrafe), realizada por la función de evaluación asociada a ella sin recurrir a funciones adicionales. Este plan consta de dos metas: *Say_Preposition* y *Say_Nominal_Group*.

La meta *Say_Preposition*, a diferencia de las otras a las que hemos hecho referencia hasta este punto de la exposición del ejemplo, no pertenece a la jerarquía *Syntax_Goal*, sino a la jerarquía *Lexical_Goal*. Como clase derivada de *Lexical_Goal* y siendo el ítem léxico asociado, *EN*, un elemento de la jerarquía *Word*, el nuevo plan es reducido a una única meta: *Say_Word*. Cabe destacar que éste es el momento en el que el sistema recupera la representación ortográfica del ítem léxico que irá asociada a esta nueva meta.

La evaluación de esta nueva meta del sistema presenta la primera palabra en pantalla: "EN". Recordemos que éste es uno de los casos básicos de este proceso recursivo de evaluación de metas. Esto es, la función de evaluación asociada a la meta *Say_Word* no genera nuevas metas que el sistema debe realizar. Por tanto, el estado de la pila al finalizar su evaluación es idéntica que al iniciar su evaluación y es como sigue:

Say_Nominal_Group
Say_Mark
Say_Verb
Say_Subject
Say_Mark

El sistema extrae de la pila la meta *Say_Nominal_Group* e inicia su evaluación. Siendo el elemento conceptual asociado al **TopicC** (i.e., **ThingC**) una subclase de *OBJECT*, la función de evaluación asociada a ésta última es ejecutada por el sistema. Para la realización de esta meta, el sistema comienza seleccionando el ítem léxico núcleo del sintagma nominal. En este caso, el elemento léxico *LUGAR* ha sido recuperado por la función de selección léxica definida en la clase *ZERO_D_LOCATION*. Una vez realizada esta operación, la función transfiere la misión

de construcción del plan al *especialista sintáctico* *Say_Nominal_Group_02* que evaluando las propiedades del objeto **ThingC** devuelve a *main-process* el siguiente plan:

Say_Determiner
Say_Premodifier
Say_Noun
Say_Postmodifier

La función de evaluación asociada a *Say_Determiner* invoca la función *Say_Determiner_01* que selecciona el elemento léxico *Article* y genera la meta *Say_Determiner*. La evaluación de esta meta es igual a la de *Say_Preposition* siendo, en este caso, la palabra "UN" el texto asociado a la meta *Say_Word* cuya función de evaluación presentará dicho texto en pantalla. Un inciso para explicar por qué el sistema ha decidido utilizar un artículo indefinido. El valor asignado a la propiedad «*identifiability*» en la definición del objeto **ThingC** es *NO*. Con esto, la representación conceptual transmite al sistema de generación que dicho objeto no hace referencia a un elemento específico del mundo que pueda ser identificado por los usuarios del lenguaje. Esta propiedad conceptual es expresada lingüísticamente mediante el uso de los artículos que en este caso es de la clase indefinido.

Antes de proseguir, repasemos el estado de la pila que cuenta, en este punto del proceso, con las siguientes metas pendientes de evaluación:

Say_Premodifier
Say_Noun
Say_Postmodifier
Say_Mark
Say_Verb
Say_Subject
Say_Mark

La meta a evaluar es ahora *Say_Premodifier*. El «tema» asociado es **Topic1**, valor del atributo «*property-ascription*» del elemento conceptual **ThingC**. Así, la función de evaluación asociada a la clase *PROPERTY_ASRIPTION* accede al elemento conceptual correspondiente al «*range*» del **Thing1** y recupera el ítem léxico asociado, ejecuta el *especialista sintáctico* *Say_Adjective_Group_01*. Dadas las características del **ThingD** (i.e., *Beautiful_Q*), el *especialista* devuelve a *main-process* la meta *Say_Adjective* con un ítem léxico asociado que para nuestro ejemplo es una instancia de la clase *HERMOSO_A*. Su valor en los atributos «*gender*» y «*number*» son asignados por la función de evaluación a partir del ítem léxico, núcleo del sintagma nominal.

Las siguientes dos metas evaluadas por el sistema pertenecen a la jerarquía *Lexical_Goal*, *Say_Adjective* y *Say_Noun*, que por medio de la meta *Say_Word* son presentadas las palabras "HERMOSO" y "LUGAR" en la pantalla.

Para la realización de la meta *Say_Posmodifier* con la entidad asociada a su «tema», **Thing3**, la función de evaluación correspondiente definida para todas las clase de la subjerarquía *TWO_PLACE_RELATION*, salvo redefinición, asigna el ítem léxico *DE* como enlace y ejecuta el *especialista sintáctico Say_Posmodifier_01*. Este *especialista* recibe de la función de evaluación el «tema» asociado al participante «range» que define **Thing3**. Así, el «tema» **TopicE** y el ítem léxico *DE* son asociados a la meta *Say_Prepositional_Group* generada por *Say_Posmodifier_01*.

El proceso de evaluación de esta meta, tal como ha sido expuesto anteriormente, da lugar a un plan compuesto por las metas *Say_Preposition* y *Say_Nominal_Group*. Con la participación de las funciones que ya hemos descrito detalladamente para la realización de estas dos clases de metas observamos la siguiente secuencia de situaciones:

Id	Estado parcial de la pila	Resultado de la evaluación	Texto a pantalla
1	<i>Say_Preposition</i> <i>Say_Nominal_Group</i>	<i>Say_Word</i>	
2	<i>Say_Word</i> <i>Say_Nominal_Group</i>		"DE"
3	<i>Say_Nominal_Group</i>	<i>Say_Determiner</i> <i>Say_Noun</i> <i>Say_Posmodifier</i>	
4	<i>Say_Determiner</i> <i>Say_Noun</i> <i>Say_Posmodifier</i>	<i>Say_Article</i>	
5	<i>Say_Article</i> <i>Say_Noun</i> <i>Say_Posmodifier</i>	<i>Say_Word</i>	
6	<i>Say_Word</i> <i>Say_Noun</i> <i>Say_Posmodifier</i>		"LAS"
7	<i>Say_Noun</i> <i>Say_Posmodifier</i>	<i>Say_Word</i>	
8	<i>Say_Word</i> <i>Say_Posmodifier</i>		"COSTAS"
9	<i>Say_Posmodifier</i>	<i>Say_Prepositional_Group</i>	
10	<i>Say_Prepositional_Group</i>	<i>Say_Preposition</i> <i>Say_Nominal_Group</i>	
11	<i>Say_Preposition</i> <i>Say_Nominal_Group</i>	<i>Say_Word</i>	
12	<i>Say_Word</i> <i>Say_Nominal_Group</i>		"DE"

En la evaluación de esta nueva meta *Say_Nominal_Group*, a diferencia de las anteriores, el *especialista sintáctico* aplicado para la elaboración del plan de realización es *Say_Nominal_Group_01* (en lugar de *Say_Nominal_Group_02*). Esta bifurcación se establece en la función de evaluación correspondiente definida para las clases derivadas de *OBJECT* observando la categoría del ítem léxico seleccionado. En este caso, el elemento léxico asociado a la clase *Italia_O*, *ITALIA*, es miembro de la subjerarquía *ProperNoun* (en lugar de *CommonNoun*).

En este punto del proceso, el estado de la pila de metas pendientes consta de los siguientes elementos:

Say_Mark
Say_Verb
Say_Subject
Say_Mark

La meta *Say_Mark* es otro caso básico del sistema que, del mismo modo que la meta *Say_Word*, presenta un texto en pantalla y no genera ningún nuevo elemento a introducir en la pila. En este ejemplo, con la evaluación de esta meta, el sistema tiene en pantalla el siguiente texto: "EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA,".

El proceso continúa sucesivamente con la evaluación de las metas pendientes y las que son generadas a partir de éstas. Para ver paso a paso el proceso que el sistema lleva a cabo para completar la oración con el texto "VIVIA UN RICO COMERCIANTE LLAMADO LANDOLFO." puede consultar el CD-ROM que acompaña a este trabajo.

4.3. Procedimiento de desarrollo

El desarrollo de las aplicaciones informáticas consta de tres fases: análisis, diseño e implementación. Conjugando las necesidades de planificación y las de visibilidad, estas fases se suceden de forma iterativa. Así, el proceso de desarrollo del sistema GEDEON se describe, básicamente, como un proceso recursivo de análisis, diseño e implementación que contempla simultáneamente los recursos y procesos implicados en la generación de oraciones en lenguaje natural.

Previamente a la construcción del sistema en sí, particularidades derivadas de la naturaleza del objeto de estudio y del contexto de realización de este trabajo han demandado una fase de preparación que, lejos de ser trivial, resulta conveniente considerar con cierto detalle. Un primer aspecto a mencionar hace referencia a una de las características de los lenguajes naturales, su capacidad representacional. Esto es, el mensaje lingüístico es un formato particular de codificación de información que los usuarios del lenguaje emplean para explicitar distintos aspectos de su conocimiento acerca del mundo. Es, por tanto, imprescindible que los sistemas artificiales de generación de lenguaje natural "conozcan" los distintos elementos que integran el dominio particular al que sus expresiones lingüísticas hacen referencia. Así, una tarea básica de preparación consiste en recopilar información acerca del dominio seleccionado que en nuestro caso está restringido a una versión particular de la "Historia del corsario Landolfo Rufolo" de Bocaccio (ver *Cuadro 4.1.*). En ella está contenida la información conceptual, léxica y gramatical que el sistema GEDEON debe "conocer" para generar expresiones lingüísticas que hagan referencia a las peripecias del pirata Landolfo.

Asimismo, resulta imprescindible establecer, previamente, un conjunto de directrices para guiar el proceso de construcción del sistema. Estas son consideraciones derivadas de las aportaciones que la psicolingüística y la inteligencia artificial han realizado sobre los procesos de codificación gramatical. Aspectos fundamentales en el diseño del sistema han sido definidos a partir del análisis racional de las investigaciones producidas en ambas disciplinas.

Finalmente, en esta fase preparativa nos familiarizamos con los conceptos básicos de los lenguajes de programación orientada a objetos y con las estrategias específicas de implementación. Posteriormente, examinamos las posibilidades de esta herramienta para la instrumentalización de nuestro diseño con la construcción de diversos sistemas pilotos muy limitados que han contribuido a su vez en la formulación de la arquitectura funcional del sistema GEDEON.

Realizadas las tareas de preparación, iniciamos el proceso de construcción del sistema identificando los recursos y procesos básicos requeridos para la generación de oraciones en el dominio especificado. El análisis léxico, sintáctico y conceptual del texto fuente nos fue proporcionando las unidades de información básicas que debían ser incorporadas al sistema. Así, las palabras y estructuras perifrásticas presentes en el texto, constituyen los elementos léxicos que componen el *lexicon* del sistema. Del mismo modo, las estructuras gramaticales observadas en el análisis sintáctico del texto conformaron la base para la definición de las

funciones que intervienen en el proceso de generación táctica de las oraciones. Por último, con la información semántica contenida en los elementos léxicos y sintácticos identificados, se establecieron las unidades conceptuales que participan en la configuración de la historia explicitada en el texto.

En un hermoso lugar de las costas de Italia, vivía un rico comerciante llamado Landolfo. Landolfo era dueño de inmensas riquezas pero su codicia era tan grande que vivía obsesionado con la idea de doblar su fortuna. Con el fin de hacer un buen negocio, compró un barco muy grande que cargó con mercancías diversas, y zarpó con él hacia Chipre. Pero al llegar allí descubrió que otros muchos barcos habían fondeado antes que él vendiendo el mismo tipo de mercancías que las que él había llevado consigo. Por esta razón, se vio obligado a vender sus mercancías a precio de liquidación y quedó prácticamente arruinado.

Como no quería regresar a su patria arruinado, decidió convertirse en pirata para recuperar el dinero perdido. Vendió su buque y con el dinero compró un barco ligero. Lo equipó adecuadamente y recorrió los mares saqueando todos los barcos que pudo. Tuvo tanto éxito en su nuevo oficio que en poco más de un año recuperó las pérdidas de sus negocios de comerciante y obtuvo una fortuna mucho mayor que la que poseía antes.

Considerando que ya tenía suficiente dinero, decidió regresar a su patria. Pero, apenas iniciado el viaje de vuelta, se levantó una tormenta tan fuerte que Landolfo se vio obligado a refugiarse en una pequeña isla. Poco después llegaron a la misma isla unos comerciantes genoveses. Al reconocer al pirata Landolfo, decidieron atacarle y hacerle su prisionero.

Al día siguiente, los comerciantes genoveses se echaron a la mar llevando a Landolfo encerrado en la bodega de uno de sus barcos. Pero de nuevo se levantó una fuerte tormenta. Los barcos fueron arrojados contra las rocas y se hicieron pedazos. Landolfo, para salvarse, se encaramó a una caja, se tendió sobre ella y, sirviéndose de los brazos para guiarla logró mantenerse a flote.

Pasó un día y una noche a la deriva, sin probar bocado y sin saber dónde se encontraba. Al fin empujado por la propia violencia del viento y convertido ya casi en una esponja, Landolfo llegó a la costa de la isla de Corfú. En aquel momento, en la playa, una mujer estaba fregando con arena sus utensilios de cocina. Al verlo, la mujer se asustó, pero viéndole tan agotado, decidió ayudarlo. Se acercó a él y lo llevó a su casa. Allí le dio masajes, un baño caliente, comidas y bebidas que lo recomfortaron. Cuando estuvo ya completamente restablecido, la mujer le devolvió la caja que le había salvado del naufragio.

Un día, mientras la mujer estaba ausente de la casa, Landolfo abrió la caja. Descubrió que estaba llena de joyas muy valiosas y que podía hacerse rico con su venta. Dio gracias a Dios por su suerte y decidió regresar a su patria. Allí vendió las joyas. Con el dinero que obtuvo por ellas consiguió una fortuna que era el doble de grande que la que tenía al principio. Envío una importante suma de dinero a la mujer de Corfú que le había sacado del mar, como pago por su ayuda. Nunca más se interesó por el comercio: dedicó su vida a vivir con todo esplendor y tranquilidad el resto de sus días.

Cuadro 4.1. — Versión de la "Historia del corsario Landolfo Rufolo" de Bocaccio utilizada como texto fuente en la construcción del sistema GEDEON.

Una vez identificadas y definidas estas unidades de información, el siguiente paso en el desarrollo del sistema consistió en la organización de estas unidades en estructuras adecuadas que permiten su integración en la arquitectura funcional diseñada en la fase previa, conformando, así, diseños pormenorizados de cada componente del sistema.

Definidos y organizados los distintos elementos que participan en la construcción de las oraciones del texto fuente, el proceso de desarrollo fue cambiando su foco de acción hacia los requerimientos propios de la herramienta de programación. Atendiendo a estas cuestiones, nuevas especificaciones, orientadas explícitamente a la implementación, fueron incorporadas en el diseño del sistema.

La implementación supuso, asimismo, sucesivas modificaciones sobre el diseño, referidas fundamentalmente a la definición de las operaciones implicadas en el proceso de generación táctica. Esto es, la actuación efectiva del sistema constituyó el indicador empírico que nos sirvió como criterio fundamental para determinar la adecuación de nuestro análisis y diseño. Este proceso recursivo de análisis, diseño e implementación, dio lugar a diversas versiones del sistema.

Las versiones 1.x. del sistema son versiones que contemplan, cada una de ellas, las nuevas consideraciones derivadas de la evaluación de la versión anterior, sin que ello suponga cambios en el diseño general. En la versión 1.3.1. alcanzamos con éxito el criterio básico exigido: generar las oraciones del texto fuente. El análisis detallado de la ejecución del sistema determinó la reformulación de la arquitectura funcional del sistema, originando las versiones 2.x. Tanto la descripción del sistema presentada en este capítulo como su evaluación han sido realizadas en base a esta nueva arquitectura.

Capítulo 5: Evaluación y validación del sistema

5.1. Introducción

No debemos olvidar que el objetivo último de este trabajo no es la construcción de un sistema de generación táctica sino la elaboración de una propuesta teórica sobre los procesos y las representaciones implicadas en la codificación gramatical. Si bien el desarrollo del sistema GEDEON constituye el componente principal de este trabajo, su interés está fundamentado en la utilidad que los sistemas de simulación suponen para el desarrollo de teorías psicológicas. Como hemos expuesto anteriormente, no todos los programas de inteligencia artificial pueden ser considerados teorías psicológicas. Su validez e implicación teórica para la psicología dependerán de su capacidad para reproducir patrones de datos similares a los obtenidos en sujetos reales.

El paralelismo entre el comportamiento del sistema de simulación y el de los sujetos reales, como ya hemos expuesto en el Capítulo 1, debe producirse a tres niveles (Fodor, 1968; García-Albea, 1981). Un primer nivel hace referencia a aspectos de la conducta observable. Así, la salida proporcionada por el sistema debe ser indistinguible de la producida por sujetos reales. En un segundo nivel, se exige del sistema la capacidad de generar conductas distintas a las ya observadas, esto es, el repertorio de conductas potenciales del sistema en cuestión. Ahora bien, la implicación de los sistemas de simulación para el desarrollo teórico de los modelos de procesamiento cognitivo se establece en base a la correspondencia entre las operaciones realizadas por el programa de simulación y los procesos efectuados por los sujetos. El grado de correspondencia entre ellas constituye un tercer nivel a considerar en el análisis comparativo entre el comportamiento del sistema y el de los sujetos reales.

En nuestro caso particular, las oraciones del texto fuente constituyen el conjunto mínimo de salidas que el sistema debe producir. Oraciones distintas a éstas sirven como muestra empírica de su conducta potencial. Asimismo, hemos definido un indicador del proceso de ejecución del sistema, conceptualmente paralelo a las pausas del habla espontánea, que constituye nuestra fuente de datos empíricos del comportamiento del sistema requeridos para el análisis comparativo de los procesos implicados.

Así, con objeto de estudiar la validez psicológica de nuestro sistema de simulación, evaluamos su comportamiento en los siguientes aspectos:

1. *La capacidad del sistema para generar las oraciones originales del texto fuente.* Este es el criterio básico de evaluación utilizado a lo largo del proceso de construcción del sistema. En este aspecto, se exige una correspondencia exacta de las oraciones generadas por el sistema con las originales.
2. *La capacidad del sistema para generar oraciones distintas a las del texto fuente.* A este respecto, realizando las modificaciones pertinentes en las representaciones de entrada, obtuvimos una muestra de salidas del sistema distintas a las ya observadas. Sin embargo, a diferencia del caso anterior, en el que los criterios de valoración de las oraciones generadas por el sistema están explícitamente definidos en el texto fuente, la valoración cualitativa de estas nuevas oraciones generadas por el sistema no es una cuestión trivial. Por ello, hemos realizado un estudio empírico con objeto de obtener indicadores objetivos para evaluar la calidad de la conducta potencial observada en el sistema. Este es el objetivo del *Estudio 1*.
3. *El grado de correspondencia entre el proceso de generación táctica planteado en el sistema y el proceso de codificación gramatical en la producción verbal de los sujetos reales.* Con el propósito de obtener indicadores empíricos sobre esta cuestión, realizamos el *Estudio 2*, en el que comparamos índices de complejidad de procesamiento registrados tanto en el sistema como en sujetos reales. Esto es, un análisis comparativo entre los valores observados en la ejecución del sistema y los datos obtenidos en un estudio previo sobre el patrón de distribución de las pausas en el habla espontánea (Shih, 1991).

5.2. Estudio 1

El objetivo de este estudio es la valoración objetiva de la calidad de las oraciones generadas por el sistema, distintas a las originales. En el aspecto cualitativo, estas oraciones no deben diferir de las producidas por sujetos reales. Para obtener un indicador empírico de esta diferencia, diseñamos una tarea de identificación de la fuente de las oraciones. Con esto,

intentamos observar el grado en que oraciones generadas por el sistema difieren de las producidas por sujetos reales en la medida en que son identificadas como tales. Esto es, la eficacia en la identificación de la fuente será mayor cuanto más pronunciadas sean las diferencias observadas entre ellas. Asimismo, cuando unas y otras sean indistinguibles, la probabilidad de acierto en la identificación de la fuente será aleatoria.

5.2.1. Método

5.2.1.1. Sujetos

Un total de 40 sujetos configuraron la muestra de este estudio. Todos ellos eran usuarios competentes del castellano (lengua materna de todos los sujetos de este estudio), con un nivel de formación mínimo de bachillerato (para una distribución de los sujetos por nivel de formación, véase *Tabla 5.1.*).

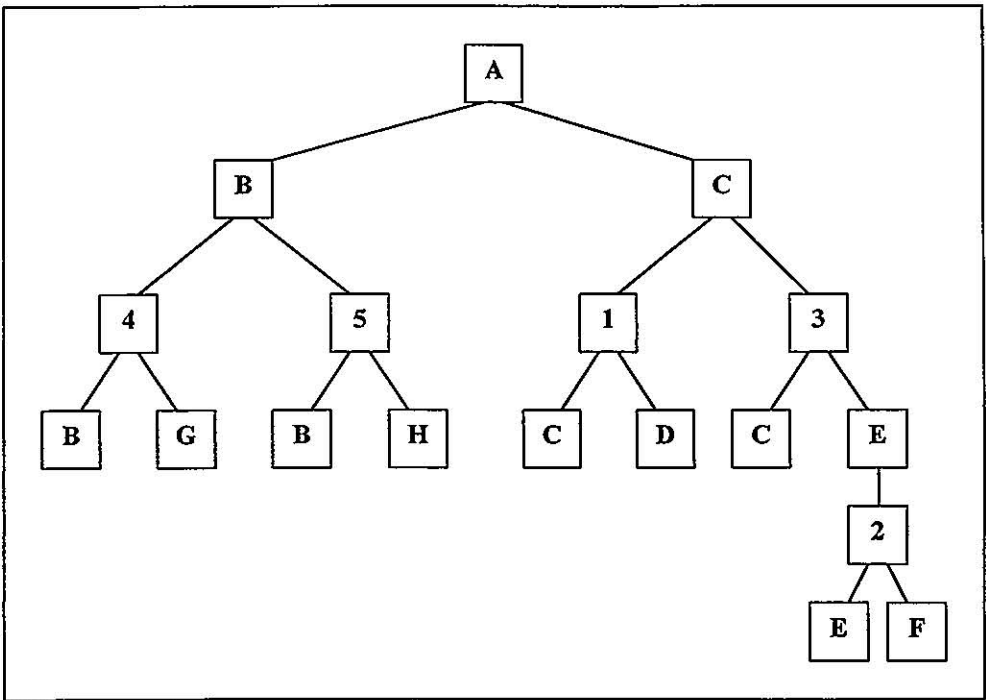
	Bachillerato	Diplomatura	Licenciatura	Doctorado
Número de sujetos	3	3	23	11

Tabla 5.1.—Número de sujetos por nivel de formación.

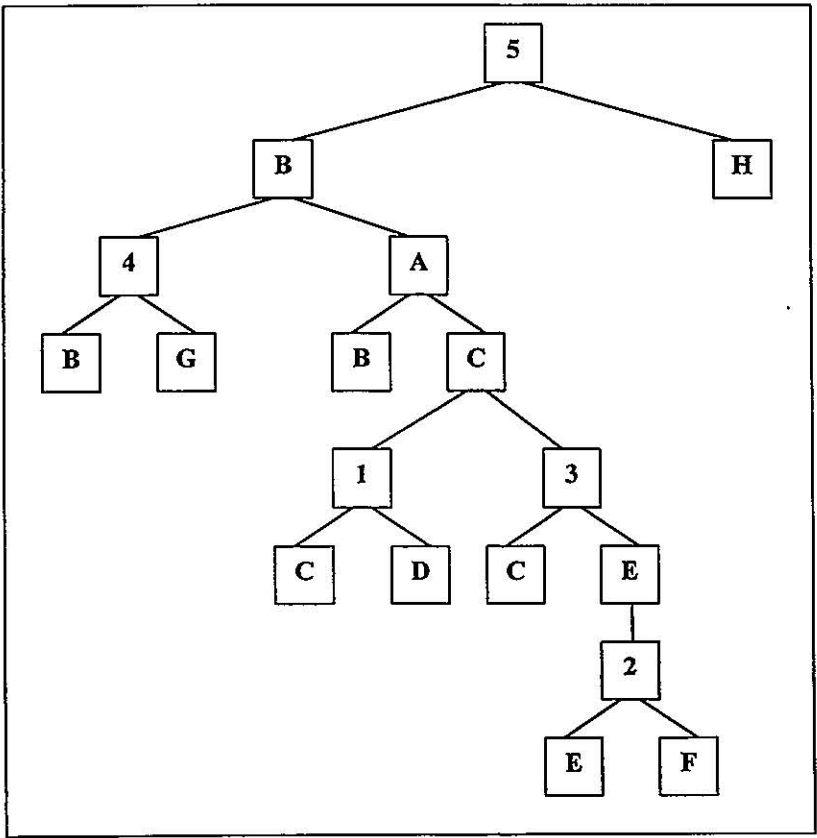
5.2.1.2. Procedimiento

A los sujetos se les presentaban una serie de oraciones y tenían que decidir si habían sido producidas por un ordenador o por una persona. Para cada uno de ellos, un subconjunto de oraciones era seleccionado de forma aleatoria de entre un total de 65 oraciones (35 generadas por el sistema y 30 por sujetos reales) con la única restricción de que entre las oraciones presentadas hubiera al menos un 30% de cada grupo.

Como ya hemos comentado, las oraciones generadas por el sistema, objeto de valoración en este estudio, se obtuvieron introduciendo algunas modificaciones sobre las propiedades de las representaciones de entrada originales. Así, por ejemplo, la oración "UN RICO COMERCIANTE QUE VIVIA EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA SE LLAMABA LANDOLFO." es generada por el sistema a partir de una representación de entrada que difiere del original, presentado en el Capítulo 4 (ver página 105), únicamente en las relaciones establecidas entre sus entidades conceptuales (ver *Cuadros 5.1. y 5.2.*).



Cuadro 5.1. — Estructura de relaciones entre las entidades conceptuales de la representación de entrada original.



Cuadro 5.2. — Nueva estructura de relaciones entre las entidades conceptuales originales.

Para la obtención de oraciones producidas por sujetos reales, pedimos a 4 personas que escribieran oraciones alternativas a las que les presentamos (texto fuente). Por tanto, a modo ilustrativo, las alternativas a la oración original en cuestión, esto es, la primera oración del texto fuente, utilizadas en este estudio son (para una consulta detallada de las oraciones utilizadas, ver *Todo GEDEON*):

Oración original:

EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA, VIVIA UN RICO COMERCIANTE LLAMADO LANDOLFO.

Oraciones alternativas generadas por el sistema:

1. EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA, VIVIA UN RICO COMERCIANTE QUE SE LLAMABA LANDOLFO.
2. LANDOLFO, UN RICO COMERCIANTE, VIVIA EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA.
3. LANDOLFO, QUE VIVIA EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA, ERA UN RICO COMERCIANTE.
4. UN RICO COMERCIANTE QUE VIVIA EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA SE LLAMABA LANDOLFO.
5. UN RICO COMERCIANTE LLAMADO LANDOLFO VIVIA EN UN HERMOSO LUGAR DE LAS COSTAS DE ITALIA.

Oraciones alternativas producidas por personas:

1. VIVIA UN RICO COMERCIANTE LLAMADO LANDOLFO EN UN PRECIOSO LUGAR EN LA COSTA ITALIANA.
2. EN UN BELLO LUGAR DE LAS COSTAS ITALIANAS VIVIA UN RICO COMERCIANTE DE NOMBRE LANDOLFO.
3. LANDOLFO ERA EL NOMBRE DE UN RICO COMERCIANTE QUE VIVIA EN UN BONITO LUGAR DE LAS COSTAS DE ITALIA.
4. EN LA COSTA DE ITALIA HABIA UN HERMOSO LUGAR EN EL CUAL VIVIA EL RICO COMERCIANTE LANDOLFO.

Cuadro 5.3.—Ejemplos de oraciones utilizadas en este estudio.

La presentación de la tarea de identificación se realizó mediante un programa informático. Tras solicitar algunos datos personales (nombre, edad, nivel de formación y lengua materna), se les presentaban las siguientes instrucciones:

"A continuación se presentarán una serie de oraciones. Deberás decidir en el menor tiempo posible si ha sido elaborada por una persona o por el ordenador. Se te presentarán todas en conjunto en una primera fase y luego podrás verlas una a una y tomar tu decisión. Recuerda hacerlo en el menor tiempo posible. Dispones de un tiempo limitado para decidir."

Finalizada la presentación de las instrucciones, las oraciones a evaluar eran presentadas de manera conjunta durante un minuto. Transcurrido este tiempo, el programa presentaba cada una de ellas en el siguiente formato:

Decide a quién pertenece la siguiente frase picando sobre el icono correspondiente.

ZONA DE PRESENTACION DE LA ORACION

Ordenador Persona

Marcador de tiempo.

Figura 5.1.—Formato de presentación de los ítems en la tarea de identificación.

Para cada ítem presentado se registró la respuesta emitida por el sujeto junto al identificador del grupo al que pertenecía dicho ítem.

5.2.1.3. Criterios de análisis

Para cada sujeto calculamos su proporción de aciertos en la identificación de oraciones generadas por el sistema (*pao*) y su proporción de aciertos en la identificación de oraciones producidas por sujetos reales (*pap*). Como indicador de la probabilidad de acierto en la identificación de la fuente, calculamos el índice de eficacia en la tarea definida como el promedio de dichas proporciones de aciertos ($idx = (pao + pap) / 2$), en el que 0'5 es su valor esperado. Esto es, una probabilidad aleatoria en la identificación correcta de la fuente de las oraciones.

5.2.2. Resultados

5.2.2.1. Análisis descriptivo

Inicialmente realizamos una serie de análisis descriptivos del índice (*idx*) con objeto de observar sus propiedades estadísticas y así determinar la prueba estadística a utilizar para el contraste de hipótesis. Los valores de los estadísticos descriptivos obtenidos están presentados en las *Tablas 5.2. y 5.3.*

<i>N Válidos</i>	<i>Media</i>	<i>Mediana</i>	<i>Moda</i>	<i>Desv. Típ.</i>	<i>Mínimo</i>	<i>Máximo</i>
40	0'5250	0'5625	0'58	0'1817	0'13	0'88

Tabla 5.2.—Estadísticos descriptivos de la variable *idx*.

<i>Asimetría</i>		<i>Curtosis</i>	
<i>Estadístico</i>	<i>Error Típico</i>	<i>Estadístico</i>	<i>Error Típico</i>
-0'324	0'374	-0'327	0'733

Tabla 5.3.—Estadísticos de distribución de la variable *idx*.

Como podemos observar, los valores de asimetría y curtosis son próximos a cero con errores típicos asociados dentro del rango $[-2, +2]$. Estos datos indican normalidad para la distribución de nuestro índice de eficacia (*idx*).

Asimismo, con el mismo propósito, realizamos la prueba de Kolmogorov-Smirnov para la comparación de la distribución muestral de nuestro índice con la distribución normal teórica. Los resultados obtenidos indican que no hay diferencias estadísticamente significativa entre ambas distribuciones ($z = 0'796$; $p = 0'550$).

Comprobada la normalidad de la distribución de la muestra, optamos por una prueba de contrastación paramétrica: la prueba T para una muestra.

5.2.2.3. Contraste de hipótesis

La evaluación cualitativa de las salidas producidas por el sistema es definida, aquí, en términos de la eficiencia de los sujetos en la identificación de la fuente de las oraciones presentadas. Así, tal como hemos expuesto al inicio de la presentación de este estudio, la probabilidad de acierto en la tarea será aleatoria cuando no se observen diferencias significativas entre las oraciones generadas por el sistema y las producidas por las personas. Esto es, el valor esperado de la media de nuestro índice de eficacia (*idx*) obtenida en la muestra es 0'5.

Realizado el análisis estadístico observamos que no había diferencias estadísticamente significativas entre la media empírica y la esperada ($T = 0'870$; $p = 0'390$). Por tanto, concluimos que no disponemos de datos empíricos suficientes para rechazar que la probabilidad de acierto en la identificación de la fuente de las oraciones es aleatoria.

5.2.3. Conclusiones

Los resultados obtenidos indican que los sujetos realizaron de forma aleatoria la tarea de identificación de la fuente de las oraciones. Esto nos permite concluir que entre ambos grupos de oraciones no se han observado suficientes diferencias cualitativas. Con esto, establecemos de forma objetiva una valoración positiva de la conducta potencial de nuestro sistema a partir de los casos que de ella se observan. Por tanto, podemos afirmar que existen indicadores empíricos a favor del sistema GEDEON en cuanto a su capacidad para generar conductas apropiadas distintas a las observadas. Esto es, el segundo criterio exigido a los sistemas de simulación.

5.3. Estudio 2

El objetivo de este segundo estudio es analizar el grado de correspondencia entre nuestro sistema de generación táctica y sujetos reales en cuanto a los procesos implicados en la construcción de oraciones. Esto es, comparar indicadores empíricos de procesamiento observados en el sistema con los observados en los sujetos reales

A diferencia de los sistemas de generación de lenguaje natural, en los sujetos reales, los procesos de producción verbal no pueden ser ejecutados paso a paso para observar las operaciones realizadas en cada momento. No obstante, la psicolingüística cuenta con distintas estrategias metodológicas para obtener datos empíricos que describan aspectos puntuales de los procesos implicados.

Una de estas estrategias utilizadas en la investigación psicolingüística de los procesos de producción verbal es el análisis de los patrones temporales del habla. Fundamentado en una lógica similar a los tiempos de reacción, los periodos de interrupción o pausas observadas en el habla espontánea son considerados indicadores de actividades de planificación durante la producción verbal (Henderson, Goldman-Eisler y Skarbeck, 1965; Goldman-Eisler, 1972; Butterworth, 1975; Holmes, 1988). Así, el análisis de la duración y el patrón de distribución de las pausas ha contribuido al esclarecimiento de cuestiones relacionadas con la naturaleza de las unidades de planificación implicadas en la construcción de mensajes verbales.

Los datos obtenidos en un estudio previo sobre las pausas en el habla (Shih, 1991) constituyeron la mitad de la información empírica requerida para la realización de este estudio. Esto es, indicadores empíricos acerca de los procesos de codificación gramatical de sujetos reales. En el trabajo mencionado se analizaron la distribución y duración de las pausas identificadas en las verbalizaciones registradas de 20 sujetos en las tareas de descripción de una lámina y de narración de una película de 4 minutos de duración previamente proyectada. Un total de 1272 pausas fueron identificadas, localizadas y medidas mediante el procedimiento establecido (no haremos referencia a aspectos de este trabajo no relevantes para el presente estudio). Cada una de ellas fueron clasificadas según la categoría sintáctica a la que precedía. Para cada una de las 19 categorías consideradas finalmente en el trabajo (ver *Tabla 5.4.*), obtuvimos la duración media de sus pausas. Estos han sido los valores considerados para nuestro análisis comparativo.

Oración	Simple Compuesta	
Inter-Cláusula	Cláusula principal	
	Cláusula yuxtapuesta	
	Cláusula coordinada	Copulativa No copulativa
	Cláusula subordinada	Adverbial No adverbial
Intra-Cláusula	Sintagma	Nominal Verbal Preposicional
	Palabra	
Alta dependencia: subordinada no adverbial		
Baja dependencia: coordinada copulativa + subordinada adverbial		

· *Tabla 5.4.: Categorías gramaticales utilizadas para la clasificación de las pausas (Shih, 1991).*

Establecido el criterio de comparación, nos enfrentamos a la difícil tarea de definir un indicador empírico del proceso de generación de nuestro sistema paralelo a las pausas del habla espontánea. Varias son las posibilidades consideradas a este respecto. En primer lugar, lógicamente, consideramos una medida de tiempo. Esto es, el tiempo transcurrido entre la emisión de una palabra y su predecesora. Sin embargo, la velocidad de ejecución de los programas de ordenador y su variabilidad en función de operaciones de entrada/salida hace de las medidas de tiempo un indicador más sensible al funcionamiento del ordenador que al sistema propiamente dicho.

Descartados los indicadores temporales optamos por alternativas conceptualmente equivalentes a la interpretación de las pausas, esto es, indicadores de la actividad de

planificación realizada por el sistema en la generación de las oraciones. Entre las distintas posibilidades consideradas, optamos por la más sencilla, a la vez menos manipulada: el número de veces que el sistema elabora un plan para alcanzar una meta determinada hasta la emisión de una palabra. Esto es, en los términos empleados en la descripción del sistema, el número de veces que el componente *main-process* recibe un plan al ejecutar la operación "evaluar meta" (consultar *Figura 4.16.*, página 104).

Sin duda, el índice definido para observar el comportamiento del sistema no es comparable con los datos de pausas en términos absolutos. Así, el grado de correspondencia del comportamiento del sistema y de los sujetos reales en cuanto a los procesos implicados será determinado por la correlación observada entre la medida y distribución de sus respectivos indicadores de actividades de planificación. En esto, el grado de correlación positiva entre ambos indicadores nos informará de la medida en que los procesos de codificación gramatical postulados en el sistema GEDEON son equivalentes a los de los sujetos reales.

5.3.1. Método

5.3.1.1. Procedimiento

Un total de 25 oraciones fueron generadas por el sistema para llevar a cabo este estudio. Se han contabilizado, para cada palabra, el número de planes elaborados por el sistema hasta su emisión. Los datos registrados fueron clasificados según la unidad de estructura lingüística a la que precedía la pausa, siguiendo las prescripciones establecidas en el estudio de las pausas al que hemos hecho referencia (ver *Tabla 5.4.*). De las 19 categorías propuestas, dos fueron eliminadas para este estudio al no observarse ningún caso correspondiente a dichas categorías (cláusulas yuxtapuesta y cláusulas no copulativas). Los valores promedios de nuestro índice planificación fueron calculados para cada una de las categorías consideradas.

5.3.1.2. Criterios de análisis

Se han definido 4 índices de planificación en este estudio. Uno correspondiente al proceso de generación de las oraciones del sistema GEDEON y tres índices de pausas registrados en el habla de sujetos reales. Por tanto, para cada una de las categorías consideradas en este estudio, contamos con un valor para cada uno de los siguientes índices:

1. Promedio del número de planes elaborados por el sistema (*idxC*)
2. Duración media en segundos de las pausas registradas en una tarea de descripción (*idxPD*)
3. Duración media en segundos de las pausas registradas en una tarea de narración (*idxPN*)
4. Promedio del tiempo de pausas en segundos entre ambas tareas ($idxP = (idxPD + idxPN) / 2$)

La correlación entre *idxC* y los índices de pausas (*idxPD*, *idxPN* y *idxP*) sería el indicador empírico que nos informará del grado de correspondencia entre los procesos de construcción de oraciones implicados en el sistema GEDEON y en los sujetos humanos.

5.3.2. Resultados

5.3.2.1. Análisis descriptivo

Realizamos un análisis descriptivo de los datos únicamente para observar el rango de los valores considerados en cada índice y señalar las diferencias existentes entre los índices referentes al sistema y a los sujetos reales. Estos resultados están resumidos en las Tablas 5.5. y 5.6.

	<i>Media</i>	<i>Mínimo</i>	<i>Máximo</i>
<i>idxC</i>	3'7134	1'49	6'50
<i>idxPD</i>	1'1669	0'51	2'73
<i>idxPN</i>	0'8014	0'52	1'10
<i>idxP</i>	0'9841	0'52	1'91

Tabla 5.5.— Estadísticos descriptivos de los índices utilizados en el estudio (resultados globales).

<i>Categoría</i>	<i>idxC</i>	<i>idxPD</i>	<i>idxPN</i>	<i>idxP</i>
Or. simple	6'50	2'73	1'10	1'91
Or. compuest	6'00	1'54	1'04	1'29
Prop. principal	5'87	1'54	1'04	1'29
Prop. copulativa	3'86	1'32	0'89	1'11
Prop. adverbial	3'43	0'80	0'65	0'73
Prop. no adverbial	3'53	1'01	0'74	0'88
Sintagma nominal	3'47	1'18	0'76	0'97
Sintagma verbal	1'49	0'58	0'72	0'65
Sintagma prepos.	3'55	0'91	0'73	0'82
Palabra	1'62	0'51	0'52	0'52
Cláusulas coord.	3'86	1'18	0'89	1'03
Cláusulas subord.	3'81	0'84	0'70	0'77
Sintagma	2'81	1'05	0'75	0'90
Interclausales	3'96	1'54	0'88	1'21
Intraclausales	2'23	1'03	0'71	0'87
Alta dependencia	3'53	1'01	0'74	0'88
Baja dependencia	3'63	1'06	0'77	0'92

Tabla 5.6.— Valor medio de los índices para cada categoría.

5.3.2.2. Contraste de hipótesis

Para contrastar la correlación existente entre nuestros índices de planificación, calculamos los correspondiente coeficientes de correlación de Spearman (ver *Tabla 5.7.*).

	<i>idxPD</i>	<i>idxPN</i>	<i>idxP</i>
<i>idxC</i>	$r = 0'794$ $p = 0'000$	$r = 0'817$ $p = 0'000$	$r = 0'839$ $p = 0'000$

Tabla 5.7.—Resultados del análisis de correlación de Spearman entre los índices del estudio.

Como podemos observar, se han encontrado correlaciones positivas estadísticamente significativas entre los índice del sistema y de los sujetos reales. Sin embargo, antes de establecer las conclusiones que de estos resultados se derivan, consideramos preciso un análisis más detallado de la relación entre los índices propuestos, ya que algunas de las categorías de clasificación eran combinaciones de otras categorías. Así, por ejemplo, la categoría sintagma integra las categorías de sintagma nominal, verbal y preposicional que, a su vez, junto a la categoría de palabra configuran una nueva categoría, la intra-cláusula.

Por ello, realizamos un nuevo análisis, eliminando las categorías combinadas por si el valor de la correlación obtenido pudiera estar artificialmente aumentado por la múltiple inclusión de los valores de los índices. Los resultados obtenidos en el análisis de correlación para las 9 categorías básicas finales son los siguientes (*Tabla 5.8.*):

	<i>idxPD</i>	<i>idxPN</i>	<i>idxP</i>
<i>idxC</i>	$r = 0'917$ $p = 0'001$	$r = 0'883$ $p = 0'002$	$r = 0'917$ $p = 0'001$

Tabla 5.8.—Resultados del nuevo análisis de correlación entre los índices del estudio.

Los resultados de este nuevo análisis no difieren del anterior en cuanto a la decisión del contraste. En ambos casos, encontramos una correlación positiva y estadísticamente significativa entre el índice de planificación del sistema y los de los sujetos reales. Esta relación entre los índices pueden ser observadas gráficamente en las *Figuras 5.2., 5.3. y 5.4.* que presentamos a continuación.

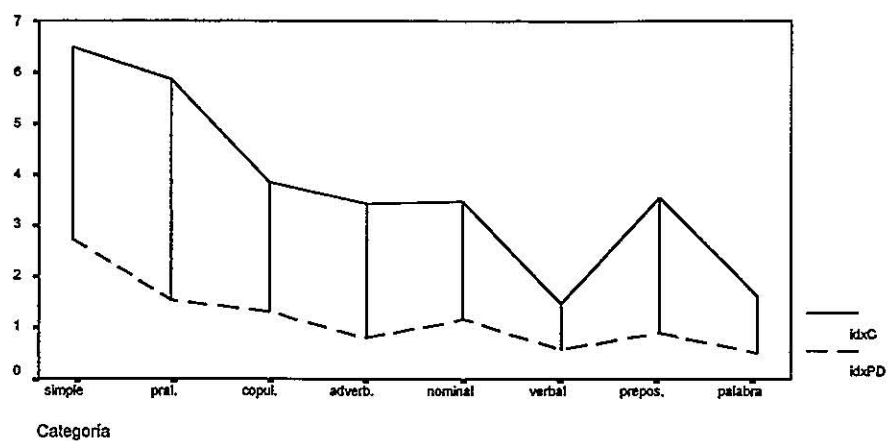


Figura 5.2.—Representación gráfica de los índices $idxC$ e $idxPD$ para las categorías básicas.

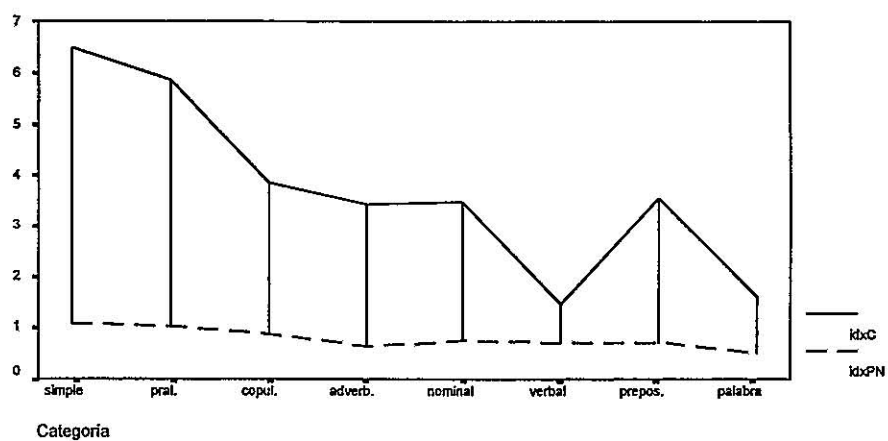


Figura 5.3.—Representación gráfica de los índices $idxC$ e $idxPN$ para las categorías básicas.

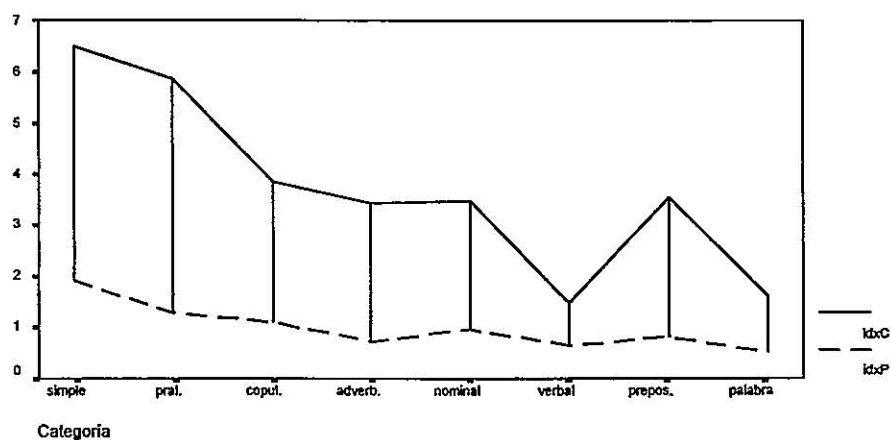


Figura 5.4.—Representación gráfica de los índices $idxC$ e $idxP$ para las categorías básicas.

Como vemos en estas gráficas, el índice de planificación del sistema se ajusta mejor a la secuencia observada en la distribución de las pausas en la tarea de descripción. Este hecho es reflejado numéricamente por sus índice de correlación, así como vimos en la *Tabla 5.8.*, el valor de correlación entre los índices *idxC* e *idxPD* es mayor que entre *idxC* e *idxPN*.

Asimismo, podemos observar en las gráficas una menor variabilidad en la distribución de las pausas en la tarea de narración (*idxPN*). Esta misma observación se deriva de los valores obtenidos en los estadísticos de dispersión (varianza y amplitud) y distribución (asimetría y curtosis), en los que el índice *idxPN* tienen valores inferiores respecto a los del índice *idxPD* (ver *Tabla 5.9.*).

	<i>Casos Válidos</i>	<i>Varianza</i>	<i>Amplitud</i>	<i>Asimetría</i>	<i>Curtosis</i>
<i>idxC</i>	9	2'7473	5'01	0'429	-0'089
<i>idxPD</i>	9	0'4499	2'21	1'716	3'603
<i>idxPN</i>	9	0'0337	0'58	0'502	-0'269
<i>idxP</i>	9	0'1760	1'40	1'455	2'510

Tabla 5.9.— Estadísticos descriptivos de los índices para las categorías básicas.

5.3.3. Conclusiones

Los resultados obtenidos muestran una relación estadísticamente significativa entre la distribución de la actividad de planificación respecto a las distintas estructuras lingüísticas observada en nuestro sistema de generación de oraciones y el patrón de distribución de las pausas registradas en el habla espontánea de sujetos reales.

Teniendo en cuenta que tanto las pausas como el índice definido para el sistema son medidas empíricas de los procesos implicados en la codificación gramatical, la relación observada nos permite afirmar que existe un alto grado de correspondencia entre el modelo de generación de oraciones implementado en el sistema GEDEON y los procesos de codificación gramatical. Sin duda, esta afirmación está sometida a las limitaciones propias de la metodología observacional en el estudio psicolingüístico, y más específicamente, las de los estudios con medidas de tiempo, así como a la propia operativización del *idxC*.

Esto es, las pausas son indicadores cuantitativos de procesamiento requerido en los distintos puntos del discurso. Sin embargo, resultan muy poco informativas acerca de las operaciones específicas realizadas en ellas. La opacidad de las pausas como indicador de los

procesos de producción verbal es manifiesta en dos aspectos observados en este estudio. En primer lugar, el uso de un índice conceptualmente similar a las pausas y, por tanto, poco informativo de las operaciones subyacentes, para el análisis de nuestro sistema, no impone la misma falta de transparencia a dichas operaciones. Así, podemos observar que índices con un mismo valor equiparan funciones de complejidad tan distintas como la asignación funcional y la recuperación de la forma ortográfica. En esto podemos advertir las limitaciones de medidas temporales como las pausas para la identificación de las operaciones participantes en los procesos de codificación gramatical. En segundo lugar, la diferencia observada en la relación entre este índice y las pausas según el tipo de tarea es otro indicador de esta opacidad. Esto es, en los discursos narrativos la presencia de la secuencia temporal es fundamental. Esta característica supone una actividad de planificación en la producción de este tipo de textos a un nivel supraoracional que no es exigido en los textos descriptivos y, obviamente, no están contemplados en un generador táctico. Nuevamente, observamos que tras las pausas se entremezclan operaciones difícilmente distinguibles en estudios basadas en medidas observacionales.

Con todo, realizados los estudios empíricos, podemos afirmar, con las reservas que imponen las limitaciones observadas en los índices utilizados, que el sistema GEDEON supera con éxito los criterios exigidos para ser considerado un sistema de simulación de una función cognitiva. Esto nos permite proponer el modelo de generación de oraciones instrumentalizado en el sistema GEDEON como un posible modelo de los procesos cognitivos implicados en la codificación gramatical.

Capítulo 6: Conclusiones generales

Una vez expuesto el trabajo fundamental realizado para la presente investigación, sólo nos queda subrayar, a modo de conclusiones de esta obra, algunas implicaciones teóricas acerca de la naturaleza de los procesos implicados en la codificación gramatical.

La propuesta que realizamos se deriva directamente de la observación del comportamiento del sistema GEDEON que, a la vista de los resultados obtenidos en los estudios de validación, puede ser considerado como un sistema de simulación de los procesos de generación de lenguaje natural. Con esto, no estamos afirmando categóricamente la veracidad de nuestras observaciones sino que, como toda propuesta teórica, habremos de someter su posterior confirmación o refutación al juicio de los datos empíricos. En esta misma línea de argumentación no queremos que se olvide que el desarrollo del sistema GEDEON ha sido fundamentalmente un ejercicio de integración de determinados aspectos teóricos y conceptuales entre dos disciplinas, la psicología y la inteligencia artificial.

Como tal práctica unificadora, la nuestra constituye apenas una propuesta racional en espera de nuevos análisis empíricos. No obstante, no debemos asimismo desatender los resultados que hemos obtenido en los estudios de validación que, pese a sus conocidas limitaciones, nos ofrecen ciertas garantías a la hora de establecer estas conclusiones.

Una vez remarcadas las salvedades pertinentes, pasamos a la exposición sucesiva de las consideraciones teóricas derivadas de nuestra investigación.

1. La codificación gramatical se puede definir, fundamentalmente, como un proceso de evaluación de metas. Esto quiere decir que, en la codificación de oraciones, un progreso continuo de elaboración de planes apropiados, permiten al sistema alcanzar su objetivo.

De esta forma, la codificación gramatical puede ser considerada como un sistema "experto" que cuenta con conocimientos lingüísticos suficientes para realizar con éxito la tarea de construir oraciones en lenguaje natural. Esta particular concepción de los procesos de codificación gramatical supone importantes implicaciones teóricas no sólo para la

psicolingüística, sino también para la psicología en general. En el punto cuarto de estas conclusiones generales volveremos a hacer referencia a esta cuestión.

2. Las funciones de evaluación asociadas a las metas o, en su caso, los *especialistas sintácticos*, son los responsables de ejecutar las operaciones de *asignación funcional*, *selección léxica*, *ensamblaje de constituyentes* y *flexión*. La participación de cada una de estas operaciones está determinada por la meta y por el elemento, conceptual o léxico, asociado a ella.

En términos generales, podríamos organizar estas operaciones en tres grupos en los que su contribución al proceso crece gradualmente en un sentido y decrece en otro, en función del ámbito gramatical considerado en cada momento.

De esta forma, la *asignación funcional* es una operación que se ejecuta con más frecuencia en funciones asociadas a unidades sintácticas clausales, mientras que la *flexión* está asociada únicamente a la evaluación de metas léxicas. Sin embargo, las operaciones de *selección léxica* y de *ensamblaje de constituyentes* conforman el grupo intermedio que está presente en la construcción tanto de unidades clausales como de unidades sintagmáticas.

3. El proceso de codificación gramatical, además de ser "dirigido por metas", es un proceso guiado por el contenido. Esto constituye un aspecto fundamental de nuestra propuesta, tanto es así que, como hemos expuesto en el capítulo cuarto dedicado a la descripción del sistema, muchas de las funciones responsables de la elaboración de planes están asociadas a las entidades conceptuales.

Una consecuencia inmediata de esta propiedad del sistema GEDEON, está reflejada en la arquitectura del programa que planteamos aquí como modelo del proceso de codificación gramatical. Esto es, a diferencia del modelo psicolingüístico de referencia (Bock y Levelt, 1994), en nuestra propuesta resulta inviable la especificación del proceso en términos de operaciones.

Como podemos observar en la *Figura 6.1.*, donde integramos nuestra propuesta con otros componentes de la producción verbal, el modelo resultante queda definido fundamentalmente por los conocimientos que intervienen. Sin embargo, es importante destacar que en ellos están incluidos aspectos tanto declarativos como procedimentales.

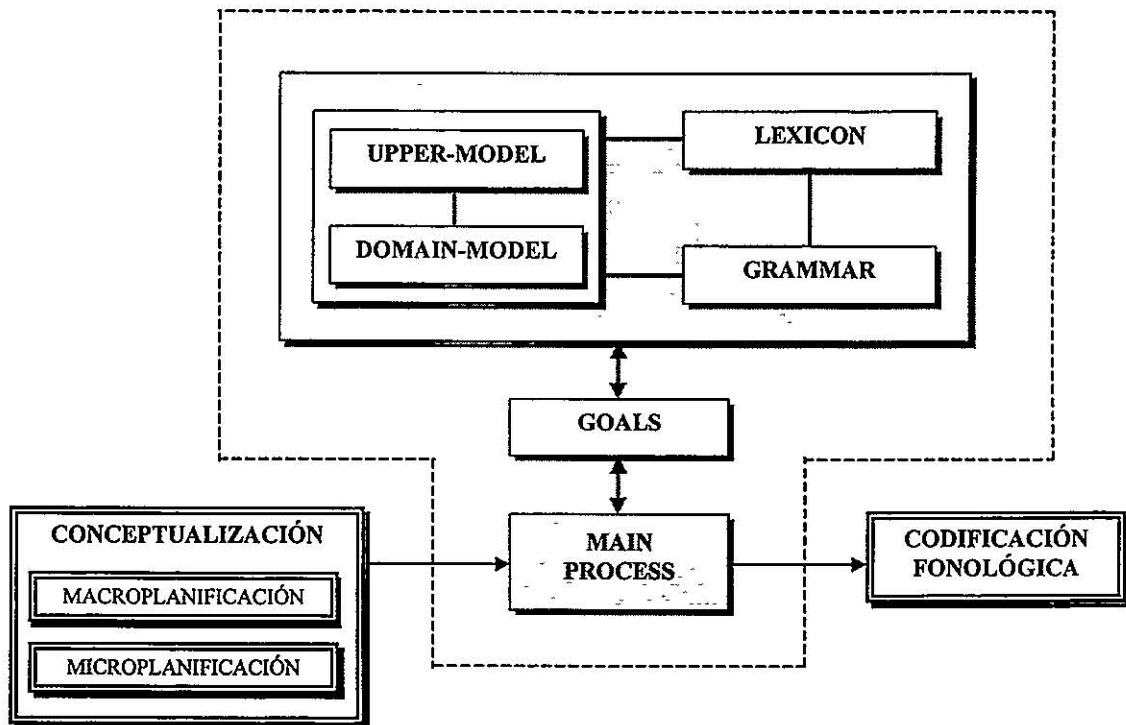


Figura 6.1.: Integración del modelo teórico propuesto en el proceso de producción verbal.

4. Las propiedades "dirigido por metas" y "guiado por contenidos" del proceso de codificación, nos permiten sugerir una concepción de la producción del lenguaje en la que un proceso central (i.e., *main-process*) gestiona la evaluación de metas que serán realizadas por los correspondientes componentes del sistema (ver *Figura 6.2.*).

Es más, con este planteamiento no sólo podemos integrar en una misma arquitectura funcional otros componentes del proceso de producción, sino también los de la comprensión verbal e incluso componentes de procesamiento no lingüísticos.

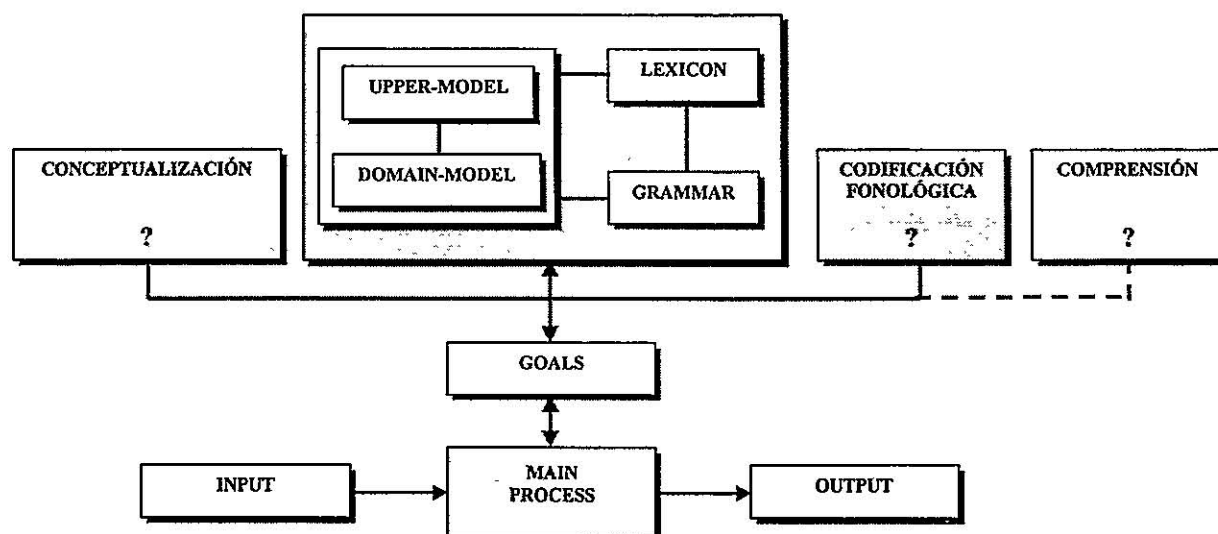


Figura 6.2.: Arquitectura integrada de los componentes del procesamiento lingüístico.

5. Por último, no podemos dejar de señalar en nuestras conclusiones, las múltiples posibilidades que la metodología de la simulación ofrece para la construcción teórica de la psicología. Con independencia de las posibles implicaciones teóricas derivadas de los sistemas de simulación, hemos comprobado en la realización de este trabajo, la capacidad potencial de esta metodología. Aún más, el propio proceso de construcción del sistema constituye una fuente de hipótesis de investigación. A continuación volveremos a insistir sobre esta cuestión en la presentación de las posibles líneas futuras de investigación derivadas de estas conclusiones.

Retomando la idea expresada en el primer párrafo de este trabajo, el desarrollo científico se fundamenta tanto en el análisis teórico como en el estudio empírico. Así, toda especulación teórica debe conducirnos hacia la investigación empírica. En nuestro caso particular, la validación empírica del sistema GEDEON constituye un aspecto clave para determinar la realidad psicológica del modelo teórico de codificación gramatical aquí propuesto. Es, por tanto, imprescindible para la continuación de este trabajo y el consecuente desarrollo de un modelo computacional del proceso de codificación gramatical, establecer una línea de investigación encaminada a la comprobación empírica de la validez de nuestras consideraciones teóricas.

De esta manera el comportamiento del sistema debe dar cuenta, en primera instancia, de los datos empíricos existentes. El estudio comparativo entre el índice de planificación del sistema y las pausas del habla espontánea, presentado en el capítulo anterior, es un ejemplo de

ello. Asimismo, el análisis del funcionamiento del sistema debe ser una fuente de generación de hipótesis para nuevas investigaciones empíricas.

Una línea hará referencia a la participación de la operación *selección léxica* durante el proceso de codificación gramatical. Esto es, planteamos como hipótesis, tras observar la ejecución del sistema, que la selección del elemento léxico específico para la expresión del elemento conceptual asignado a la función de sujeto gramatical, a diferencia de otros elementos conceptuales, es ejecutada por *especialistas sintácticos* del ámbito proposicional (interclausal) inmediatamente después de seleccionar el verbo núcleo de la proposición.

Otra vía de continuación de este trabajo está directamente relacionada con la comprobación empírica de la validez psicológica del sistema. Los resultados obtenidos en estas investigaciones son indicadores de aciertos y errores presentes en nuestro modelo. El futuro desarrollo del sistema GEDEON como modelo teórico, estará orientada, fundamentalmente, a la incorporación de modificaciones prescritas por las investigaciones empíricas.

Sin duda, no está en nuestras intenciones mantener el sistema GEDEON en el estado actual. Así, además de superar sus limitaciones como modelo explicativo del proceso de codificación gramatical, planteamos como objetivo de futuro inmediato ampliar su base de conocimiento. Esto es, introducir nuevos elementos conceptuales y léxicos al sistema y definir nuevos *especialistas sintácticos* y criterios de decisión (i.e., criterios pragmáticos).

Nuestro proyecto futuro no se limita únicamente a "hacer crecer" el sistema de generación de oraciones. Verlo desarrollarse no será más que una submeta del plan elaborado tras la evaluación de la meta inicial: construir el sistema GALENA (Generador Automático de LEnguaje NATural). Esto es, construir un modelo de los procesos cognitivos implicados en la producción verbal.

En fin, soñar no cuesta nada.

Referencias bibliográficas

- Adarraga, P. (1991). El marco general de la psicología cognitiva. En C. Castilla del Pino y J.M. Ruiz Vargas (Eds.), *Aspectos cognitivos de la esquizofrenia*. Madrid: Trotta.
- Adarraga, P. (1993). Memoria de oposición no publicada. Universidad Autónoma de Madrid.
- Adarraga, P. (1994). El marco de la ciencia cognitiva. En P. Adarraga y J.L. Zaccagnini (Eds.), *Psicología e inteligencia artificial*. Madrid: Trotta.
- Allen, J. (1989). Natural language understanding. En A. Barr, P.R. Cohen, y E.A. Feigenbaum (Eds.), *The handbook of artificial intelligence, Vol. 4*. Reading, MA: Addison-Wesley.
- Anderson, J.R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Bateman, J.A., Kasper, R.T., Moore, J.D., y Whitney, R.A. (1990). A general organization of knowledge for natural language processing: The PENMAN upper model. USC/Information Sciences Institute Technical Report.
- Belinchón, M., Riviére, A. e Igoa, J.M. (1992). *Psicología del lenguaje. Investigación y teoría*. Madrid: Trotta.
- Bar-Hillel, Y. (1960). The present status of automatic translation of languages. En F.L. Alt (Ed.), *Advances in computers*. Nueva York: Academic Press.
- Barr, A., y Feigenbaum, E.A. (1981). Understanding natural language. En A. Barr y E.A. Feigenbaum (Eds.), *The handbook of artificial intelligence, Vol. 1*. Los Altos, CA: William Kaufmann.
- Beardon, C., Lumsden, D., y Holmes, G. (1991). *Natural language and computational linguistics. An introduction*. Nueva York: Ellis Horwood.
- Bock, K. (1986). Meaning, sound and syntax: Lexical priming in sentence production. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 12, 575-586.
- Bock, K. (1987). Co-ordinating words and syntax in speech plans. En A.W. Ellis (Ed.), *Progress in the psychology of language, Vol. 3*. Londres: Lawrence Erlbaum Associates.
- Bock, K., y Levelt, W.J.M. (1994). Language production: Grammatical encoding. En M. Gernsbacher (Ed.), *Handbook of psycholinguistics*. San Diego, CA: Academic Press.
- Boden, M.A. (1977). *Artificial intelligence and natural man*. Nueva York: Basic Books. [Trad. Cast. en Madrid: Tecnos, 1984.]
- Booth, A.D. (1948). Translating machines. *International Social Science Bulletin*, 10, 55-62.
- Bresnan, J. (1978). A realistic transformational grammar. En M. Halle, J. Bresnan y G.A. Miller (Eds.), *Linguistic theory and psychological reality*. Cambridge, MA: MIT Press.

- Bresnan, J. (1981). An approach to universal grammar and the mental representation of language. *Cognition*, 10, 39-52.
- Bresnan, J., y Kaplan, R. (1982). Lexical-functional grammar: A formal system for grammatical representation. En J. Bresnan (Ed.), *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.
- Blumenthal, A.L. (1970). *Language and Psychology: Historical aspects of psycholinguistics*. Nueva York: Wiley.
- Butterworth, B. (1975). Hesitation and grammatical encoding. *Language and Speech*, 8, 148-158.
- Butterworth, B. (1980a). Introduction: A brief review of methods of studying language production. En B. Butterworth (Ed.), *Language production, Vol. 1. Speech and talk*. Londres: Academic Press.
- Butterworth, B. (1980b). Evidence from pauses in speech. En B. Butterworth (Ed.), *Language production, Vol. 1. Speech and talk*. Londres: Academic Press. [Trad. Cast. en F. Valle, F. Cuetos, J.M. Igoa y S. del Viso (Eds.), *Lecturas de psicolingüística, Vol 1. Comprensión y producción del lenguaje*. Madrid: Alianza, 1990.]
- Butterworth, B. (1982). Speech errors: Old data in search of new theories. En A. Cutler (Ed.), *Slips of the tongue and language production*. Berlín: Mouton.
- Cawsey, A. (1998). *The essence of artificial intelligence*. Englewood Cliffs, NJ: Prentice-Hall.
- Collins, A., y Smith, E.E. (Eds.) (1988). *Readings in cognitive science: A perspective from psychology and artificial intelligence*. San Mateo, CA: Morgan Kaufmann.
- Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2, 113-124.
- Chomsky, N. (1957). *Syntactic Structures*. La Haya: Mouton.
- Chomsky, N. (1959a). A review of verbal behavior, by B.F. Skinner. *Language*, 35, 26-58. [Trad.Cast. en R. Bayés (Ed.), *¿Chomsky o Skinner? La génesis del lenguaje*. Barcelona: Fontanella, 1977.]
- Chomsky, N. (1959b). On certain formal properties of grammars. *Information and Control*, 2.
- Dale, R., Mellish, C., y Zock, M. (Eds.) (1990). *Current research in natural language generation*. Londres: Academic Press.
- Danlos, L. (1983). Some issues in generation from a semantic representation. *Proceedings of 8th IJCAI Conference*, 606-609.
- Danlos, L. (1987). A French and English syntactic componente for generation. En G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology, and linguistics*. Dordrecht: Martinus Nijhoff.
- Dell, G.S., y Burger, L.K. (1997). Language production and serial order: A functional analysis and a model. *Psychological Review*, 104, 123-147.

- Fodor, J.A. (1968). *Psychological explanation: An introduction of the philosophy of psychology*. Nueva York: Random House. [Trad. Cast. en Madrid: Cátedra, 1980.]
- Fodor, J.A. (1981). El problema cuerpo-mente. *Investigación y Ciencia*, 54, 65-75.
- Fodor, J.A., Bever, T.G., y Garrett, M.F. (1974). *The psychology of language: An introduction to psycholinguistics and generative grammar*. Nueva York: McGraw Hill.
- Ford, M. (1982). Sentence planning unit: Implications for the speakers representation of meaningful relations underlying sentences. En J. Bresnan (Ed.), *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.
- Fromkin, V.A. (Ed.) (1973). *Speech errors as linguistic evidence*. La Haya: Mouton.
- Fromkin, V.A. (1988). Grammatical aspects of speech errors. En F.J. Newmeyer (Ed.), *Linguistics: The Cambridge survey, Vol. 2. Linguistic theory: Extensions and implications*. Cambridge: Cambridge University Press.
- García-Albea, J.E. (1981). La simulación como explicación. En L. Jáñez (Ed.), *Simulación en psicología*. Publicaciones del Departamento de Psicología Matemática. Universidad Complutense de Madrid. [Reimpreso en J.E. García-Albea, *Mente y conducta*. Madrid: Trotta, 1993.]
- Gardner, H. (1985). *The mind's new science: A history of the cognitive revolution*. Nueva York: Basic Books. [Trad. Cast. en Buenos Aires: Paidós, 1987.]
- Garrett, M.F. (1975). The analysis of sentence production. En G. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory, Vol. 9*. Nueva York: Academic Press.
- Garrett, M.F. (1980). Levels of processing in sentence production. En B. Butterworth (Ed.), *Language production, Vol. 1. Speech and talk*. Londres: Academic Press. [Trad. Cast. en F. Valle, F. Cuetos, J.M. Igoa y S. del Viso (Eds.), *Lecturas de psicolingüística, Vol 1. Comprensión y producción del lenguaje*. Madrid: Alianza, 1990.]
- Garrett, M.F. (1988). Processes in language production. En F.J. Newmeyer (Ed.), *Linguistics: The Cambridge survey, Vol. 3. Language: Psychological and biological aspects*. Cambridge: Cambridge University Press.
- Garrett, M.F. (1991). Errors and their relevance for models of language production. En G. Blanken, J. Dittman, H. Grim, J. Marshall y C. Wallesch (Eds.), *Linguistic disorders and pathologies*. Berlín: Walter de Gruyter (en prensa).
- Gazdar, G. (1981). On syntactic categories. *Philosophical Transactions of the Royal Society of London*, 259, 267-283.
- Gazdar, G., Klein, E., Pullum, G., y Sag, I.A. (1985). *Generalized phrase structure grammar*. Cambridge, MA: Harvard University Press.
- Glaser, W.R., y Döngelhoff, F.J. (1984). The time course of picture-word interference. *Journal of Experimental Psychology: Human Perception and Performance*, 10, 640-654.

- Goldman, N. (1975). Conceptual generation. En R.C. Schank (Ed.), *Conceptual information processing*. Amsterdam: North-Holland.
- Goldman-Eisler, F. (1961). A comparative study of two hesitation phenomena. *Language and Speech*, 4, 18-26.
- Goldman-Eisler, F. (1967). Sequential temporal patterns and cognitive processes in speech. *Language and Speech*, 10, 112-132.
- Goldman-Eisler, F. (1972). Pauses, clauses, sentence. *Language and Speech*, 10, 112-132.
- Grosz, B.J., Sparck Jones, K., y Webber, B.L. (Eds.) (1986). *Readings in Natural Language Processing*. Los Altos, CA: Morgan Kaufmann.
- Halliday, M.A.K. (1961). Categories of the theory of grammar. *Word*, 17, 241-292.
- Halliday, M.A.K. (1978). *Language as social semiotic*. London: Edward Arnold.
- Halliday, M.A.K. (1985). Systemic background. En J.D. Benson y W.S. Greaves (Eds.), *Systemic perspectives on discourse, Vol. 1*. Norwood, NJ: Ablex.
- Hawkins, P.R. (1971). The syntactic location of hesitation pauses. *Language and Speech*, 14, 277-288.
- Henderson, A., Goldman-Eisler, F., y Skarbeck, A. (1965). Temporal patterns of cognitive activity and breath control in speech. *Language and Speech*, 8, 236-242.
- Henderson, A., Goldman-Eisler, F., y Skarbeck, A. (1966). Sequential temporal patterns in spontaneous speech. *Language and Speech*, 9, 207-216.
- Holmes, V.M. (1988). Hesitations and sentence planning. *Language and Cognitive Processes*, 3, 323-361.
- Hovy, E.H. (1988). *Generanting natural language under pragmatic constraints*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Igoa, J.M. (1996). The relationship between conceptualization and formulation processes in sentence production: Some evidence from Spanish. En M. Carreiras, J.E. García-Albea y N. Sebastián (Eds.), *Language processing in Spanish*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Igoa, J.M., y García-Albea, J.E. (1999). Unidades de planificación y niveles de procesamiento en la producción del lenguaje. (en prensa)
- Jacobs, P.S. (1985). PHRED: A generator for natural language interfaces. *American Journal of Computational Linguistics*, 11, 219-242.
- Jacobs, P.S. (1987). KING: A knowledge-intensive natural language generator. En G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology, and linguistics*. Dordrecht: Martinus Nijhoff.
- Kay, M. (1979). Functional grammar. *Proceedings of the 5th Annual Meeting of the Berkeley Linguistics Society*, 89-95.

- Kay, M. (1985). Parsing in functional unification grammar. En D.R. Dowty, L. Karttunen y A.M. Zwicky (Eds.), *Natural Language Parsing*. Cambridge, MA: Cambridge University Press.
- Kempen, G. (Ed.) (1987). *Natural Language Generation: New results in artificial intelligence, psychology, and linguistics*. Dordrecht: Martinus Nijhoff.
- Levelt, W.J.M. (1989). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.
- MacKay, H., y Osgood, C. (1959). Hesitation phenomena in spontaneous English speech. *Word*, 15, 19-44.
- McKeown, K.R. (1982). *Generating natural language text in response to questions about database structure*. Tesis Doctoral. Universidad de Pensilvania.
- McKeown, K.R. (1985). *Text generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge, MA: Cambridge University Press.
- McKeown, K.R., Elhadad, M., Fukumoto, Y., Lim, J., Lombardi, C., Robin, J., y Smadja, F. (1990). Natural language generation in COMET. En R. Dale, C. Mellish y M. Zock (Eds.), *Current research in natural language generation*. Londres: Academic Press.
- Mann, W.C., y Matthiesen, C.M.I.M. (1983). NIGEL: A systemic grammar for text generation. USC/Information Sciences Institute Technical Report RR-83-105.
- Mann, W.C., y Matthiesen, C.M.I.M. (1985). Demonstration of the NIGEL text generation computer program. En J.D. Benson y W.S. Greaves (Eds.), *Systemic perspectives on discourse, Vol. 1*. Norwood, NJ: Ablex.
- Mann, W.C., y Thompson, S.A. (1987a). Rhetorical structure theory: Description and construction of text structures. En G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology, and linguistics*. Dordrecht: Martinus Nijhoff.
- Mann, W.C., y Thompson, S.A. (1987b). *Rhetorical structure theory: A theory of text organization*. USC/Information Sciences Institute Research Report RR-87-190.
- Mann, W.C., y Thompson, S.A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8, 243-281.
- Miranda Podadera, L. (1984). *Análisis gramatical. Curso superior de gramática española*. 38ª Edición. Madrid: Hernando.
- Navarro, J.M., y Calvo, T. (1983). *Historia de la filosofía*. Madrid: Anaya.
- Newell, A. (1980). Physical symbol systems. *Cognitive Science*, 4, 135-183.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Newell, A., Shaw, J.C., y Simon, H.A. (1964). Elements of a theory of human problem-solving. En R.J.C. Harper, C.C. Anderson, C.M. Christensen y S.M. Hunka (Eds.), *The cognitive processes: Readings*. Englewood Cliffs, NJ: Prentice-Hall. [Reimpreso en *Psychological Review*, 65, 151-166.]

- Newell, A., y Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Newell, A., y Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19, 113-126. [Reimpreso en J. Haugeland (Ed.), *Mind design: Philosophy, psychology, artificial intelligence*. Montgomery: Bradford Books, 1981.]
- Oettinger, A.G. (1957). Account identification for automatic data processing. *Journal of the ACM*, 4, 245-253.
- PENMAN (1989). PENMAN documentation: the Primer, the User Guide, and the Reference Manual. USC/Information Sciences Institute Technical Report.
- Pollard, C.J., y Sag, I.A. (1987). Information-based syntax and semantics. *CLSI Lecture Notes 13*. Chicago: University of Chicago Press.
- Pollard, C.J., y Sag, I.A. (1994). *Head-driven phrase structure grammar*. Chicago: University of Chicago Press.
- Pylyshyn, Z.W. (1984). *Computation and cognition: Toward a foundation for cognitive science*. Cambridge, MA: MIT Press. [Trad. Cast. en Madrid: Debate, 1988.]
- Pylyshyn, Z.W. (1989). Computing in cognitive science. En M.I. Posner (Ed.) *Foundations of cognitive science*. Cambridge, MA: MIT Press.
- Reifler, E. (1954). The first conference on mechanical translation. *Mechanical Translation*, 1, 23-32.
- Reiter, E. (1994). Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? *Proceedings of the Seventh International Workshop on Natural Language Generation*, 163-170.
- Rich, E., y Knight, K. (1991). *Artificial intelligence*. Nueva York: McGraw-Hill. [Trad. Cast. en Madrid: McGraw-Hill, 1994.]
- Roelofs, A. (1992). A spreading-activation theory of lemma retrieval in speaking. *Cognition*, 42, 107-142.
- Schank, R.C. (1975). *Conceptual information processing*. Amsterdam: North-Holland.
- Schlesinger, I.M. (1977). Components of a production model. En S. Rosenberg (Ed.), *Sentence production: Developments in research and theory*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Schriefers, H., Meyer, A.S., y Levelt, W.J.M. (1990). Exploring the time course of lexical access in language production: Picture-word interference studies. *Journal of Memory and Language*, 29, 86-102.
- Shannon, C.E.A. (1948). Mathematical theory of communication. *Bell System Technical Journal*, 27, 479-523.

- de Smedt, K.J.M.J. (1990). IPF: An incremental parallel formulator. En R. Dale, C. Mellish y M. Zock (Eds.), *Current research in natural language generation*. Londres: Academic Press.
- Thompson, H.S. (1977). Strategy and tactics: A model for language production. *Proceedings of the 13th Annual Meeting of the Chicago*, 89-95.
- de Vega, M. (1982). La metáfora del ordenador: implicaciones y límites. En I. Delclaux y J. Seoane (Eds.), *Psicología cognitiva y procesamiento de la información*. Madrid: Pirámide.
- del Viso, S. (1990). *Errores espontáneos del habla y producción del lenguaje*. Tesis doctoral. Editorial de la Universidad Complutense de Madrid.
- Weaver, W. (1955). Translation (1949). En W.N. Locke y A.D. Booth (Eds.), *Machine translation of languages: Fourteen essays*. Cambridge, MA: MIT Press.
- Weizenbaum, J. (1966). ELIZA: A computer program for the study of natural language communication between man and machine. *Communication of the ACM*, 9, 36-44.
- Wiener, N. (1948). *Cybernetics*. Nueva York: Wiley. [Trad. Cast. en Madrid: Guadiana de Publicaciones, 1971.]
- Woods, W. (1970). Transition network grammars for natural language analysis. *Communication of the ACM*, 13, 591-606. [Reimpreso en B.J. Grosz, K. Sparck Jones y B.L. Webber (Eds.), *Readings in natural language processing*. Los Altos, CA: Morgan Kaufmann, 1986.]
- Woods, W. (1973). Progress in natural language understanding: An application to LUNAR geology. En *Proceedings of the AFIPS Conference 42*. AFIPS Press.
- Yngve, V.H.A. (1960). A model and a hypothesis for language structure. En *Proceedings of the American Philosophical Society*, 104, 444-466.
- Zaccagnini, J.L. (1994). Introducción al campo de la inteligencia artificial. En P. Adarraga y J.L. Zaccagnini (Eds.), *Psicología e inteligencia artificial*. Madrid: Trotta.
- Zaccagnini, J.L., y Delclaux, I. (1982). Psicología cognitiva y procesamiento de la información. En I. Delclaux y J. Seoane (Eds.), *Psicología cognitiva y procesamiento de la información*. Madrid: Pirámide.

ANEXO

GEDEON

Indice de contenido

- ❑ A.1. ¿Qué es *TODO GEDEON*?
- ❑ A.2. ¿Cómo utilizar *TODO GEDEON*?
- ❑ A.3. Estructura de *TODO GEDEON*
- ❑ A.4. Interfaz del sistema GEDEON
- ❑ A.5. Recogida de Oraciones
- ❑ A.6. Validación del Sistema
- ❑ A.7. Tutorial de GEDEON

A.1. ¿Qué es *TODO GEDEON*?

El propósito de este CD-ROM es, además de recopilar gran parte de la información en formato digital que ha sido utilizada o generada como resultado de esta tesis doctoral, ejemplificar varios aspectos de esta investigación mediante la utilización de técnicas de hipermedia que por su alto poder expresivo ayudan a la mejor comprensión de los mismos.

Aún cuando el contenido ha sido elaborado empleando novedosas técnicas informáticas como son el hipertexto y la multimedia no es necesario contar con conocimientos informáticos de alto nivel para su utilización. Basta con un poco de práctica en el uso del ratón y no tener temores para la experimentación informática.

Seguramente usted, en más de una ocasión, ha requerido de alguna información que ha encontrado en Internet, o por simple curiosidad ha pasado algunas horas frente al ordenador navegando por la red de Redes. Si es su caso, cualquiera de esas horas le habrá servido de práctica para la utilización de este CD-ROM, en caso contrario, no se preocupe, no es algo muy difícil de aprender. Al utilizar el lenguaje HTML, el lenguaje empleado en Internet para la representación y distribución de la información, el entorno de trabajo de nuestro CD-ROM es muy similar a cualquiera de las páginas que habrá visitado durante sus horas de navegante (véase *figura 1*). Por supuesto que para ello tendrá que contar con algunos requerimientos mínimos que pasaremos a enumerar en la próxima sección.

A.2. ¿Cómo utilizar *TODO GEDEON*?

Para sacar algún provecho a este CD-ROM será necesario contar con un ordenador y, por supuesto, no cualquier ordenador puede sernos útil. Listemos lo mínimo que debe cumplir un ordenador tanto de hardware como de software.

Requerimientos de hardware:

1. Procesador 486 (Pentium recomendado)
2. 16 Mb de RAM
3. Tarjeta de vídeo con capacidad para resolución de 800x600 con 256 colores
4. Tarjeta de sonido con altavoces
5. Lector CD-ROM de doble velocidad.

Requerimientos de software:

1. Sistema operativo Windows 95, Windows NT o Windows 98.
2. Navegador de Internet (Netscape, Internet Explorer, etc.)

Una vez seguros de que nuestro ordenador cumple los requerimientos antes mencionados podemos ya comenzar a trabajar. Para facilitar el comienzo a los que no están adaptados al trabajo con CD-ROM listaremos el conjunto de pasos iniciales para ejecutar la aplicación principal:

Pasos para la ejecución de la aplicación principal de *TODO GEDEON*:

1. Introduzca el CD-ROM en su unidad lectora. Éste está preparado para que una vez introducido en el ordenador, comience automáticamente la aplicación principal contenida en el mismo. Los siguientes pasos sólo se realizarán en caso de que por algún motivo la aplicación principal no se ejecute automáticamente.
2. Haga clic sobre el botón *Inicio* de la *Barra de Tareas*
3. Seleccione *Ejecutar* desde el menú *Inicio* de la *Barra de Tareas*
4. En la caja de Diálogo *Ejecutar* escriba *[Unidad de CD-ROM]:\indice.html* donde *[Unidad de CD-ROM]* se refiere a la letra asignada a su lectora de CD-ROM.
5. Haga clic sobre el botón *Aceptar*

Inmediatamente se ejecutará el navegador de Internet que tenga por defecto instalado en su ordenador con la página de índice de contenidos (véase *figura 1*). Veamos en la siguiente sección cada uno de los temas contemplados en el CD-ROM.

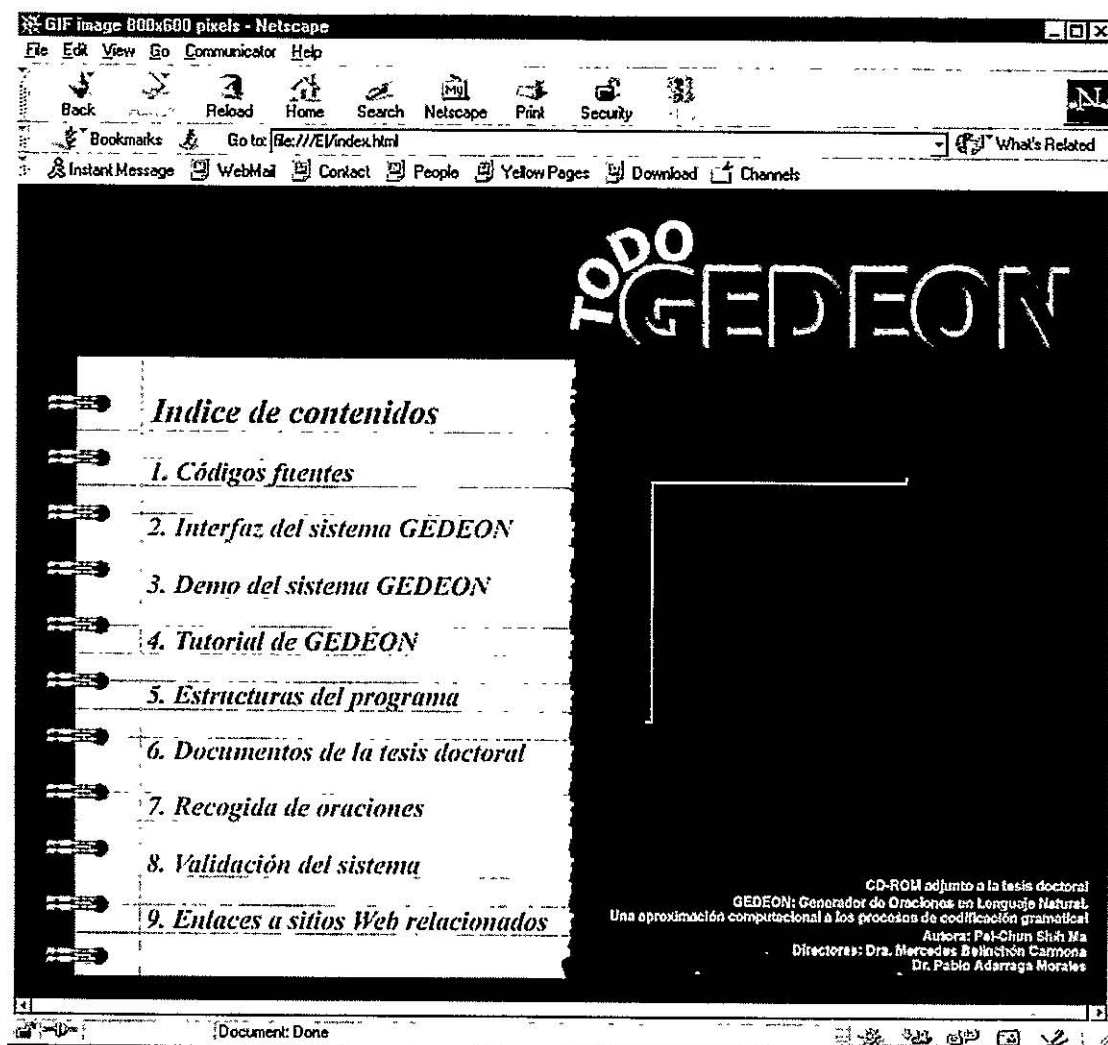


Figura 1: Pantalla Índice de Contenidos.

A.3. Estructura de *TODO GEDEON*

TODO GEDEON está formado por un conjunto de 9 apartados relacionados con el sistema GEDEON, resultado de la investigación de la que se ocupa esta tesis doctoral, que incluyen desde el código fuente de las aplicaciones generadas hasta listados de sitios Web que resultan de interés para los investigadores interesados en profundizar en aspectos relacionados con la generación de lenguaje natural (véase *figura 1*).

Apartado 1: Códigos Fuente

En este apartado hemos reunido el código fuente de las principales aplicaciones que ha sido necesario programar para la investigación realizada. Ha sido incluido en el CD-ROM por si pudiera ser de utilidad para los investigadores con conocimientos de programación que

necesiten un punto de partida para futuros trabajos de investigación relacionados con este tema. Contiene:

- Sistema GEDEON (Borland C++)
- Interfaz del sistema GEDEON (Visual Basic 5.0)
- Aplicación de recogida de datos para el sistema GEDEON (Visual Basic 5.0)
- Aplicación de validación del sistema GEDEON (Visual Basic 5.0)

Apartado 2: Interfaz del Sistema GEDEON

Se incluye la aplicación de interfaz que muestra paso a paso la ejecución del sistema GEDEON, no sólo a nivel del resultado final (oración generada) sino que en cada momento podemos observar el estado en el que se encuentra el sistema y las operaciones que se van realizando para la realización de las metas.

Apartado 3: Demo del Sistema GEDEON

El principal objetivo de este apartado es el de apoyar la explicación dada en el capítulo 4 de la tesis doctoral sobre el proceso de ejecución de GEDEON a través de un ejemplo concreto. La Demo contenida no es más que la Interfaz a la que se hace alusión en el apartado anterior pero limitada a una entrada concreta, la misma que fue utilizada en el ejemplo del capítulo 4. Haciendo clic sobre este apartado se ejecutará automáticamente la aplicación *Demo*.

Apartado 4: Tutorial de GEDEON

Este apartado contiene una aplicación desarrollada en entorno multimedia para ilustrar mediante la utilización de textos, imágenes y animaciones todos los aspectos relacionados con el diseño del sistema de generación táctica, la arquitectura del programa informático y la esencia del funcionamiento del mismo, conteniendo además la ejecución de un ejemplo que permite ilustrar los tres aspectos antes mencionados.

Apartado 5: Estructuras del Programa

Un conjunto de gráficas es el principal contenido de este apartado. Dichas gráficas ilustran el Modelo de Jerarquía de Objetos utilizadas en el diseño del sistema GEDEON. Recordemos que esta metodología ha sido la utilizada para la instrumentación de nuestro sistema y ha sido explicada con detalle en el capítulo 2 de esta tesis.

Apartado 6: Documentos de la Tesis Doctoral

Aquí se podrá encontrar una recopilación de todos los documentos en formato electrónico que conforman la tesis doctoral. Incluye todos los capítulos y anexos. Para visualizar alguno de estos documentos sólo deberá pulsar sobre este apartado y seleccionar el que desee del listado de documentos disponibles.

Apartado 7: Recogida de Oraciones

Se incluye el programa informático que fue aplicado a algunos sujetos de la investigación para que elaboraran oraciones alternativas a las originales del texto fuente, que posteriormente serían utilizadas en el proceso de validación del sistema.

Apartado 8: Validación del Sistema

Contiene la aplicación de validación del sistema que integra el conjunto de oraciones generadas por el sistema GEDEON y las oraciones generadas por los sujetos expuestos a la aplicación anterior y somete a otra muestra de personas a una tarea de identificación de las fuentes de dichas oraciones, tarea que corresponde con el estudio 1 de la investigación realizada.

Apartado 9: Enlaces a Sitios Web Relacionados

Este último apartado contiene un listado de sitios disponibles en Internet, que han sido visitados a lo largo del desarrollo de esta investigación y que consideramos de interés para aquellas personas con una línea de trabajo similar al objeto de estudio de nuestra tesis. Pulsando sobre este apartado se mostrará un listado de dichos sitios que podrán ser visitados haciendo clic sobre cualquiera de ellos.

A.4. Interfaz del sistema GEDEON

Uno de los objetivos primordiales de este trabajo fue la elaboración de una propuesta teórica sobre los procesos y las representaciones implicadas en la codificación gramatical. Para ello se hacía necesario contar con una herramienta que fuese capaz no sólo de obtener los resultados de dichos procesos, sino que además, y como en muchas herramientas de simulación, expusiera con detalle cada una de las metas necesarias de alcanzar y los diferentes estados por los que pasa el sistema a lo largo del proceso de generación. Para satisfacer este último requerimiento fue creada la aplicación que sirve de interfaz a GEDEON. Veamos a continuación cómo utilizarla.

Para ejecutar la aplicación de interfaz, una vez mostrada la pantalla de índice de contenidos del CD-ROM, pulse sobre el apartado 2 que aparece en la pantalla bajo el título *Interfaz del Sistema GEDEON* (véase figura 2).

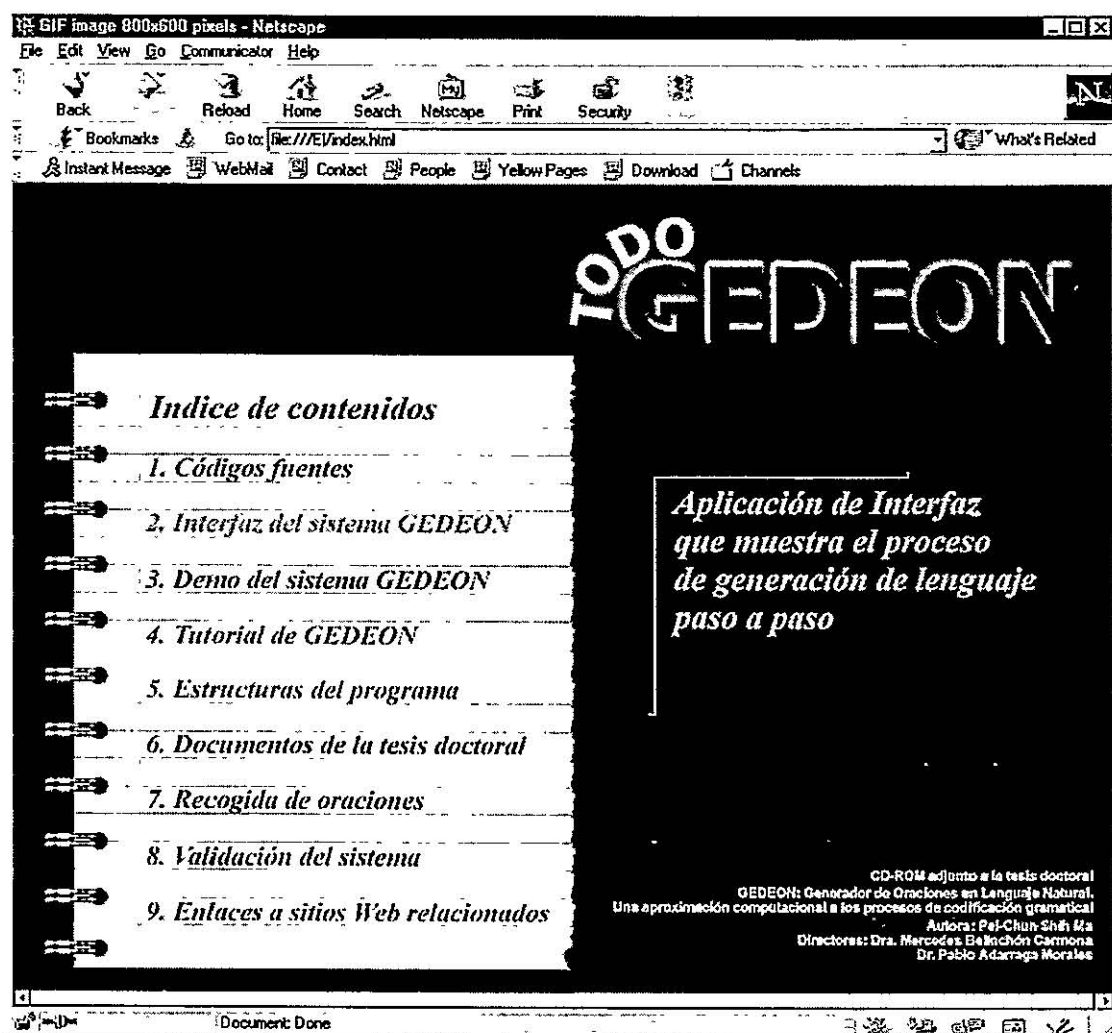


Figura 2: Pantalla Índice de Contenidos con el apartado 2 señalado.

Seleccionando esta opción se mostrará la pantalla inicial de la interfaz cuya principal funcionalidad es la de mostrar el conjunto de archivos de representaciones iniciales disponibles y donde usted podrá seleccionar el que desee como entrada al sistema. Dentro del recuadro nombrado *Available Inputs* se listarán dichas representaciones y haciendo clic sobre las mismas podrá observar la representación correspondiente (véase capítulo 4) contenida por dicho archivo en el recuadro nombrado *Input Representation* (véase figura 3).

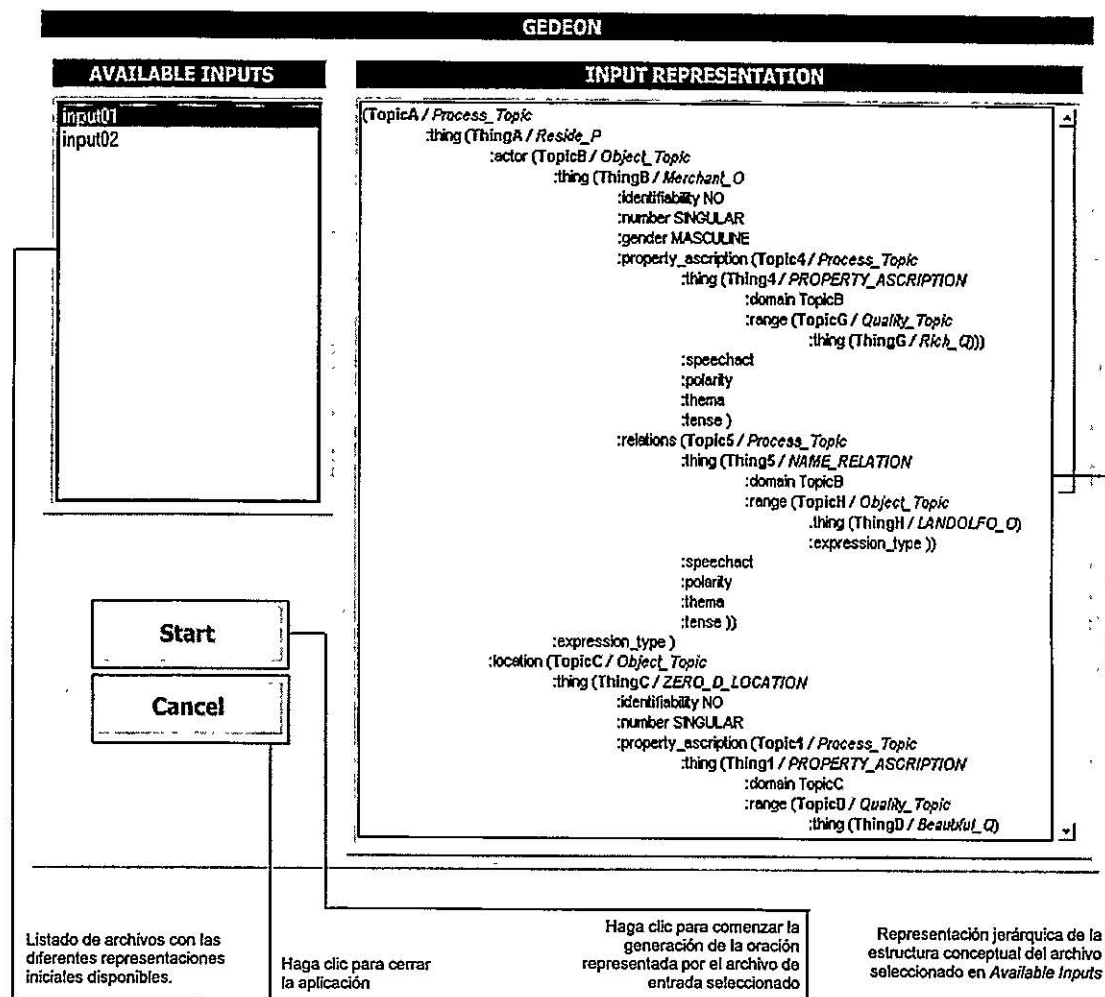


Figura 3. Pantalla inicial de Interfaz del sistema GEDEON.

Una vez seleccionado el archivo deseado podremos escoger ahora dos caminos: comenzar la generación de la oración basada en la entrada seleccionada haciendo clic sobre el botón *Start* o cancelar la ejecución de la aplicación utilizando el botón *Cancel* (véase figura 3).

Si se comienza con la generación de la oración de la entrada seleccionada, se nos muestra un nuevo entorno de trabajo, llamado *entorno de simulación*. En el mismo se muestran varios cuadros de textos con funciones específicas dentro del proceso de simulación (véase figura 4).

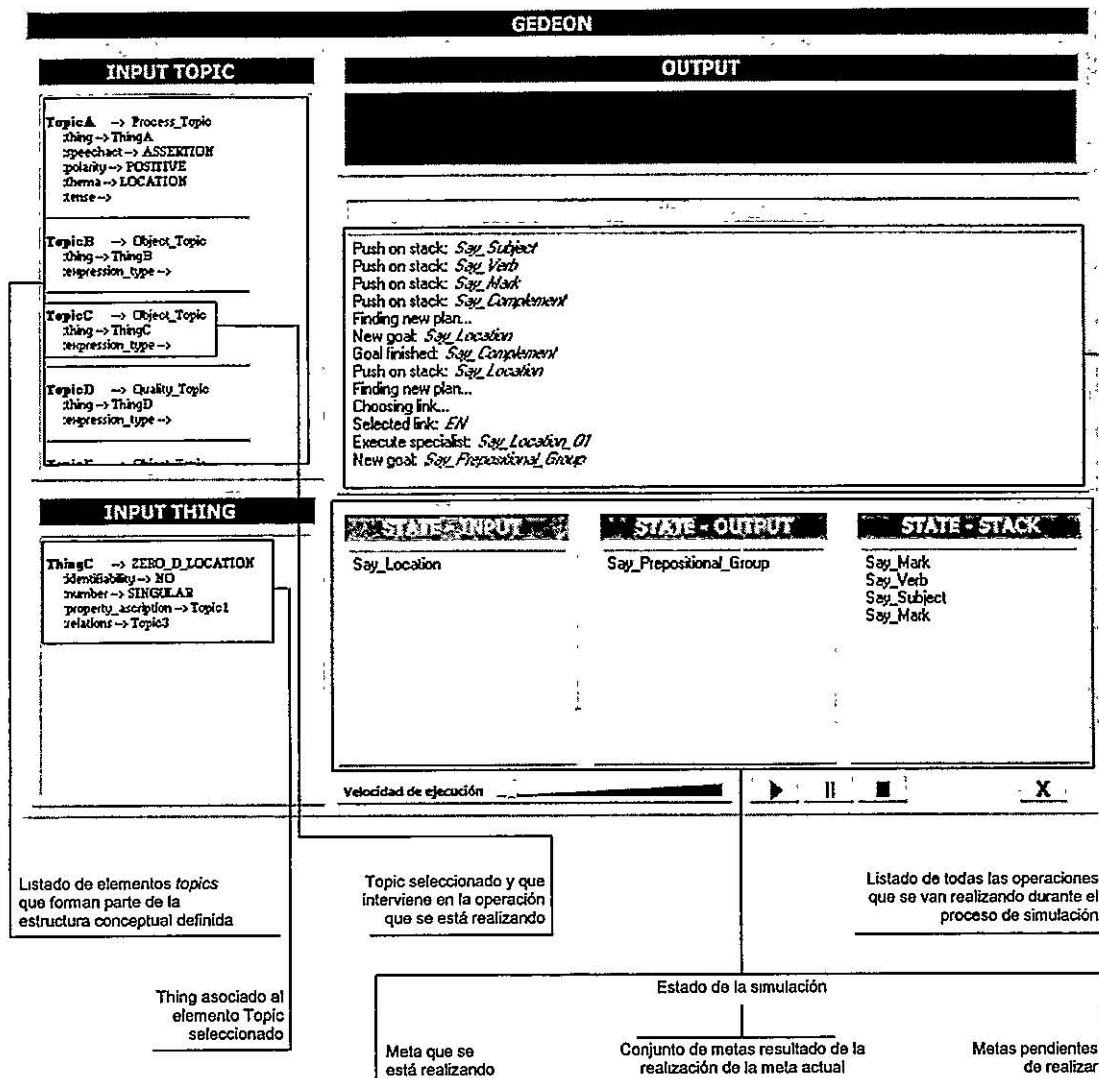


Figura 4. Entorno de simulación de Interfaz del sistema GEDEON

Este entorno es interactivo, es decir, las operaciones que se realizan pueden ser controladas hasta cierto punto por el usuario de la aplicación. Para ello se pueden emplear los botones que aparecen en la parte inferior derecha de la pantalla y que detallaremos a continuación (véase figura 5).

- El botón *Play* nos permite iniciar la ejecución del proceso de simulación.
- El botón *Pause* nos permite detener la ejecución del proceso, por ejemplo, en caso de que deseemos analizar el conjunto de metas pendientes en el *State-Stack*, las operaciones que se han realizado y que han quedado registradas en el cuadro nombrado *Operations*, o las metas que han salido como resultado de la finalización

de una meta anterior en el *State-Output*. Una vez pulsado el botón *Pause* y detenida la ejecución podrá continuar la ejecución volviendo a pulsar sobre el mismo botón.

- El botón *Stop* que permite detener la ejecución de la simulación e inicializar el estado de todos los componentes del sistema.
- El botón *Close* permite cerrar la aplicación de la interfaz.

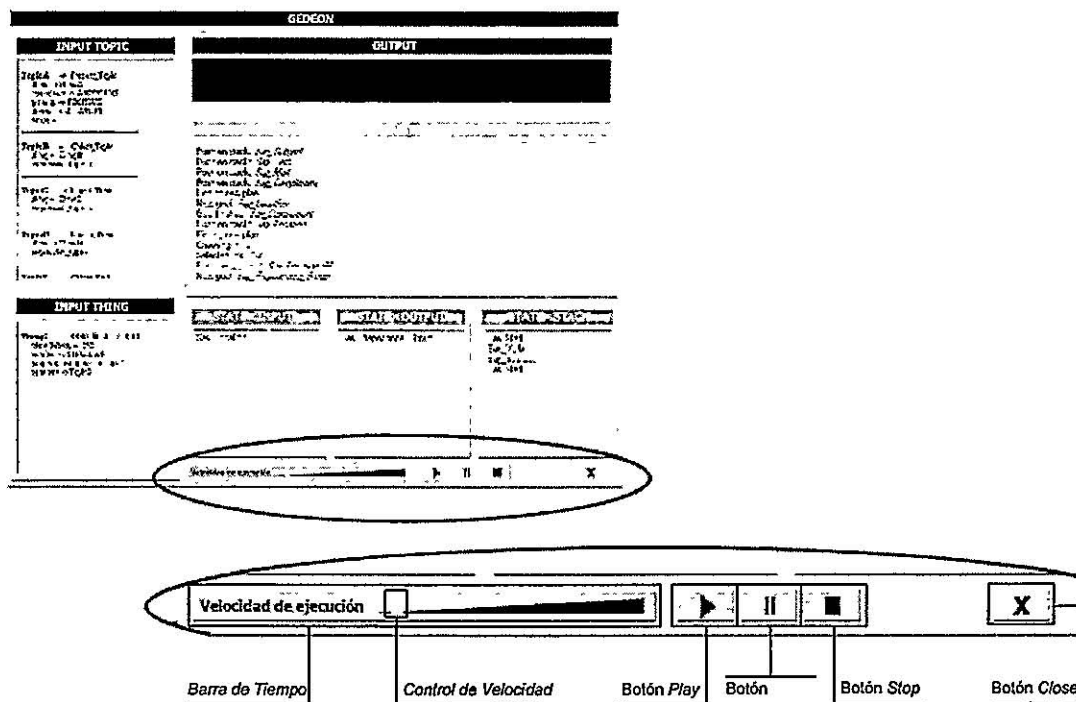


Figura 5. Barra de Botones de la Interfaz de GEDEON

Un recurso especial que se ha agregado es el control encargado de modificar la velocidad de ejecución entre paso y paso dentro de la simulación. Evidentemente su utilidad radica en la posibilidad de aumentar o disminuir la rapidez del proceso permitiendo al usuario del sistema ralentizar la ejecución y apreciar así con más claridad los pasos dentro de la realización de cada meta o acelerarla si así fuese necesario (véase figura 5).

Para la modificación de la velocidad será necesario utilizar lo que en informática se ha denominado *arrastrar y soltar* (*drag and drop*).

1. Coloque el ratón sobre el *medidor* que marca la velocidad establecida (por defecto aparecerá a la mitad de la *barra de velocidad*)
2. Presione el botón del ratón y sin liberarlo arrastre el ratón hasta que el *medidor* aparezca en la posición adecuada a la velocidad deseada

3. Libere el botón del ratón

Una vez realizado esto la simulación se ajustará al nuevo valor de tiempo seleccionado.

A.5. Recogida de Oraciones

Para realizar la validación del sistema era necesario contar con diferentes versiones generadas por personas de las oraciones originales. Con tal motivo se elaboró una aplicación que en esencia muestra un conjunto de oraciones al usuario y da la posibilidad de que éste genere hasta 10 alternativas para cada de ellas.

Para ejecutar esta aplicación haga clic sobre el apartado 7 que aparece en la pantalla de índice de contenidos, *Recogida de Oraciones* (véase figura 6).

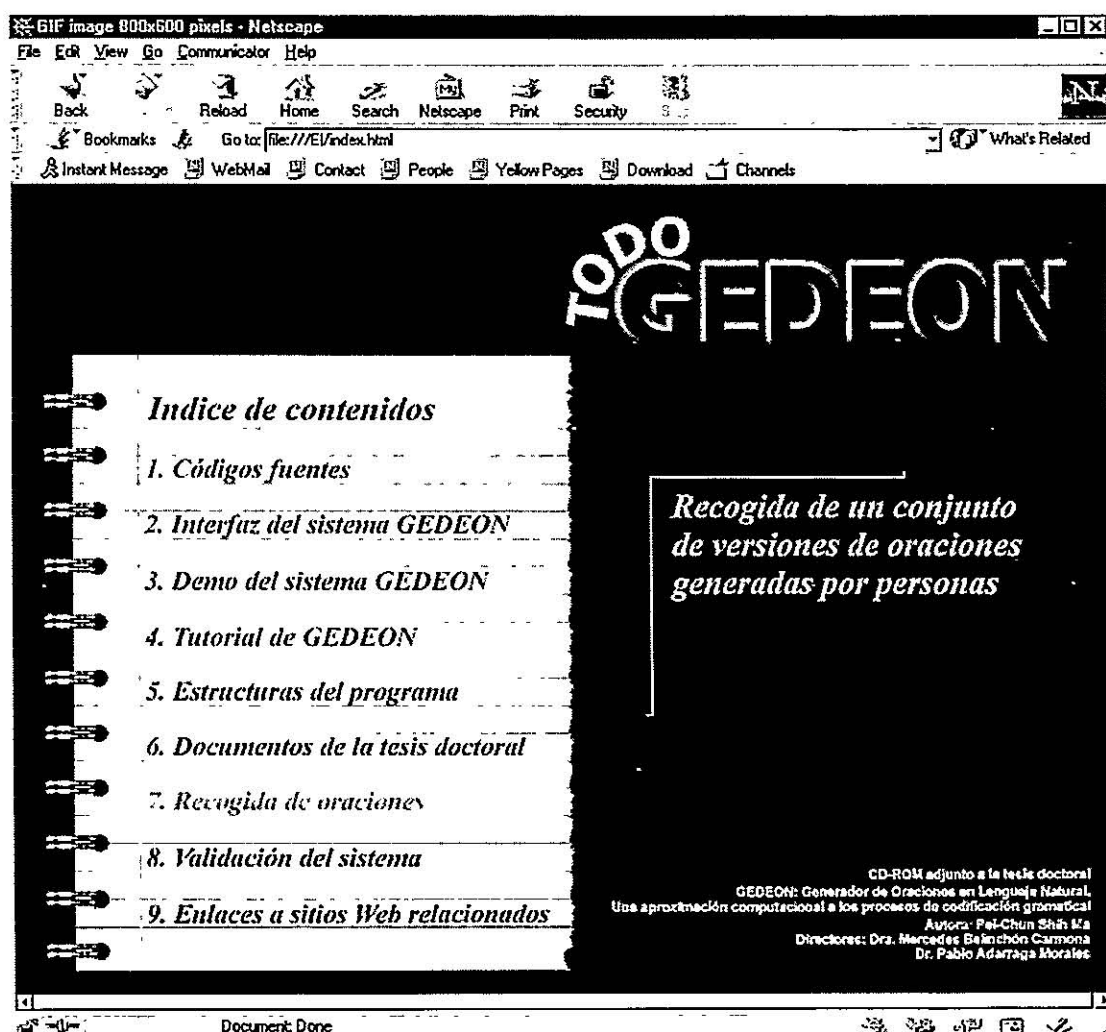



Figura 6: Pantalla Indice de Contenidos con el apartado 7 señalado.

Se muestra en pantalla la página inicial del programa que se encarga de recoger los datos de identificación del usuario (véase *figura 7*). Una vez introducidos los datos al pulsar sobre la flecha derecha pasamos a la siguiente pantalla en la que se dan las instrucciones necesarias a seguir con el resto de la aplicación.

Datos Personales

Introduzca sus datos personales en la casilla correspondiente:

Nombre:	Roberto
Apellidos:	Cabrera González
Edad:	28
Estudios:	Doctorado
Lengua materna:	castellano
Otra:	



Cuadros de texto para introducir los datos personales del usuario

Icono de Flecha Derecha para continuar la ejecución de la aplicación

Figura 7: Pantalla de recogida de datos personales, Aplicación de Recogida de Oraciones

A continuación se mostrará un conjunto de pantallas, cada una de ellas presentando la oración original correspondiente y 10 cuadros de texto donde el usuario podrá redactar sus propias versiones (véase *figura 8*).

Escriba al menos cinco oraciones alternativas a:

Como no quería regresar a su patria arruinado, decidió convertirse en pirata para recuperar el dinero perdido.

1.- Decidió convertirse en pirata para recuperar el dinero perdido y no regresar arruinado a su patria.

2.- Para recuperar el dinero perdido y no regresar arruinado a su patria, decidió convertirse en pirata.

3.-

4.-

5.-

6.-

7.-

8.-

9.-

10.-



Oración original

Cuadros de texto donde el usuario podrá elaborar entre 5 y 10 versiones de la oración original

Icono de Flecha Derecha para continuar a la próxima oración o finalizar la aplicación

Figura 8: Pantalla típica de elaboración de versiones de la oración original. Aplicación de Recogida de Datos

Una vez elaboradas las versiones de la oración original se podrá pasar a la siguiente pulsando sobre la flecha que aparece en la esquina inferior derecha hasta llegar a la última en la que la aplicación finalizará automáticamente.

A.6. Validación del Sistema

Para la validación se sometió a una muestra de personas a una aplicación que retomaba las versiones generadas por personas y por GEDEON, las mezclaba, las exponía y solicitaba al usuario que identificase la fuente de las mismas. Se intentaba demostrar que las versiones generadas por nuestro sistema GEDEON eran indistinguibles de las generadas por personas.

Si desea ejecutar la aplicación, haga clic sobre el apartado 8 titulado *Validación del Sistema* (véase figura 9).

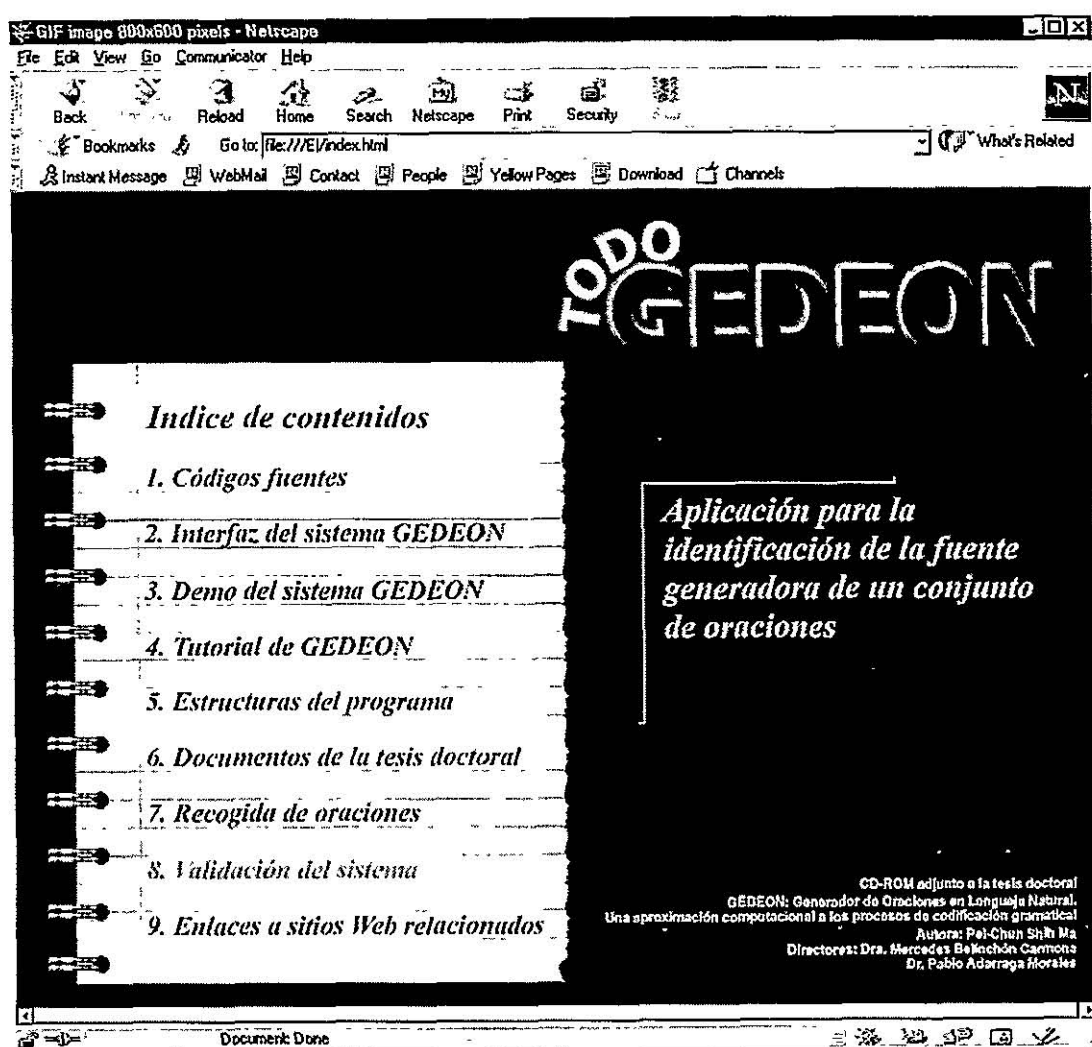


Figura 9: Pantalla Indice de Contenidos con el apartado 8 señalado.

Inicialmente, y al igual que la aplicación de recogida de datos, la pantalla inicial solicita los datos personales del usuario que deberán ser registrados (véase figura 10).

Datos Personales

Introduzca sus datos personales en la casilla correspondiente:

Nombre: Roberto

Apellidos: Cabrera González

Edad: 28

Estudios: Doctorado

Lengua materna: castellano

Otra:

Cuadros de texto para introducir los datos personales del usuario

Icono de Flecha Derecha para continuar la aplicación

Figura 10: Pantalla inicial de Aplicación de Validación.

Rellenados los datos solicitados y haciendo clic sobre el icono de la flecha situado en la parte inferior derecha de la pantalla se prosigue con la pantalla de instrucciones. Posteriormente, se muestra el conjunto de todas las oraciones que serán presentadas al usuario durante un tiempo limitado, representado por la *Barra de Progreso de Tiempo* que se muestra en la parte inferior de la pantalla (véase figura 11).

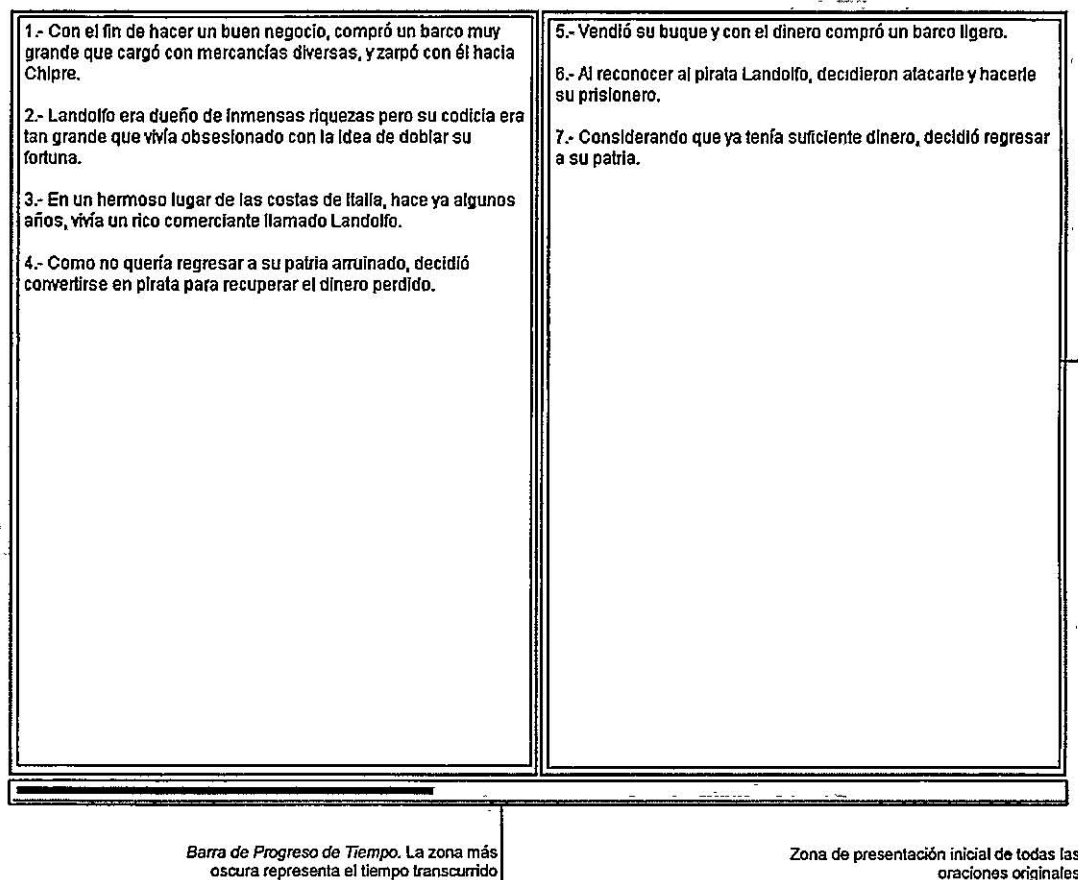


Figura 10: Pantalla de presentación inicial de las oraciones originales. Aplicación de Validación.

Pasado el tiempo se comienzan a presentar una a una las oraciones, también con un tiempo limitado durante el que el usuario deberá decidir si la oración mostrada ha sido generada por una persona o por el ordenador. Una vez que se tenga la respuesta se pulsa sobre el botón correspondiente (véase figura 11). Cuando se emite la respuesta o si el tiempo disponible ha sido consumido se pasará automáticamente a la próxima oración.

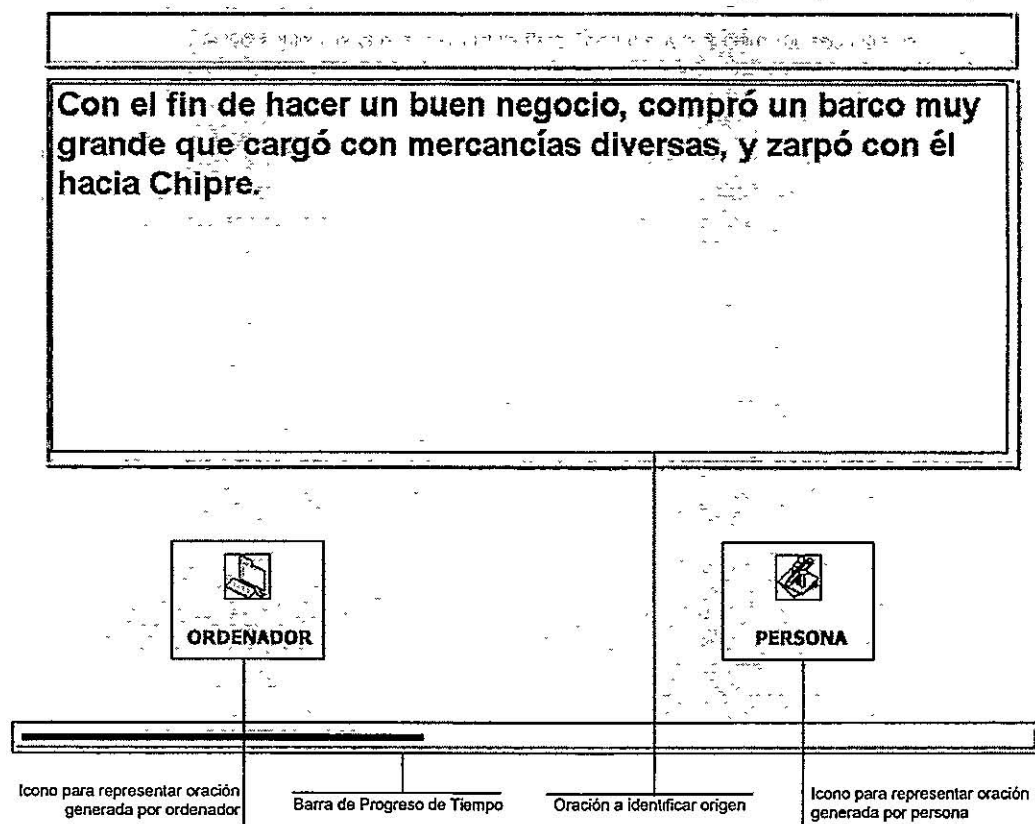


Figura 11: Pantalla para la identificación de origen de la oración. Aplicación de Validación.

A.7. Tutorial de GEDEON

Con ánimo de facilitar en gran medida la comprensión del diseño, procedimiento de desarrollo y funcionamiento del sistema en cuestión se ha confeccionado una aplicación que gracias a la integración de textos, imágenes, animaciones y sonidos (integración que ha dado en llamarse multimedia) permite ilustrar claramente qué componentes integran GEDEON, el procedimiento tomado como base para su construcción y el proceso de generación que sigue durante la ejecución del mismo.

Como muchos programas informáticos que pertenecen a la categoría de tutoriales multimedia, el nuestro consiste en un entorno interactivo en el que el usuario dispone de un

conjunto de temas vinculados todos por un hilo común pero que pueden ser consultados con independencia.

Para ejecutar el tutorial es suficiente con hacer clic sobre el apartado 4 de la página *Indice de Contenidos* de nuestro CD, identificado como *Tutorial de GEDEON* (véase figura 12).

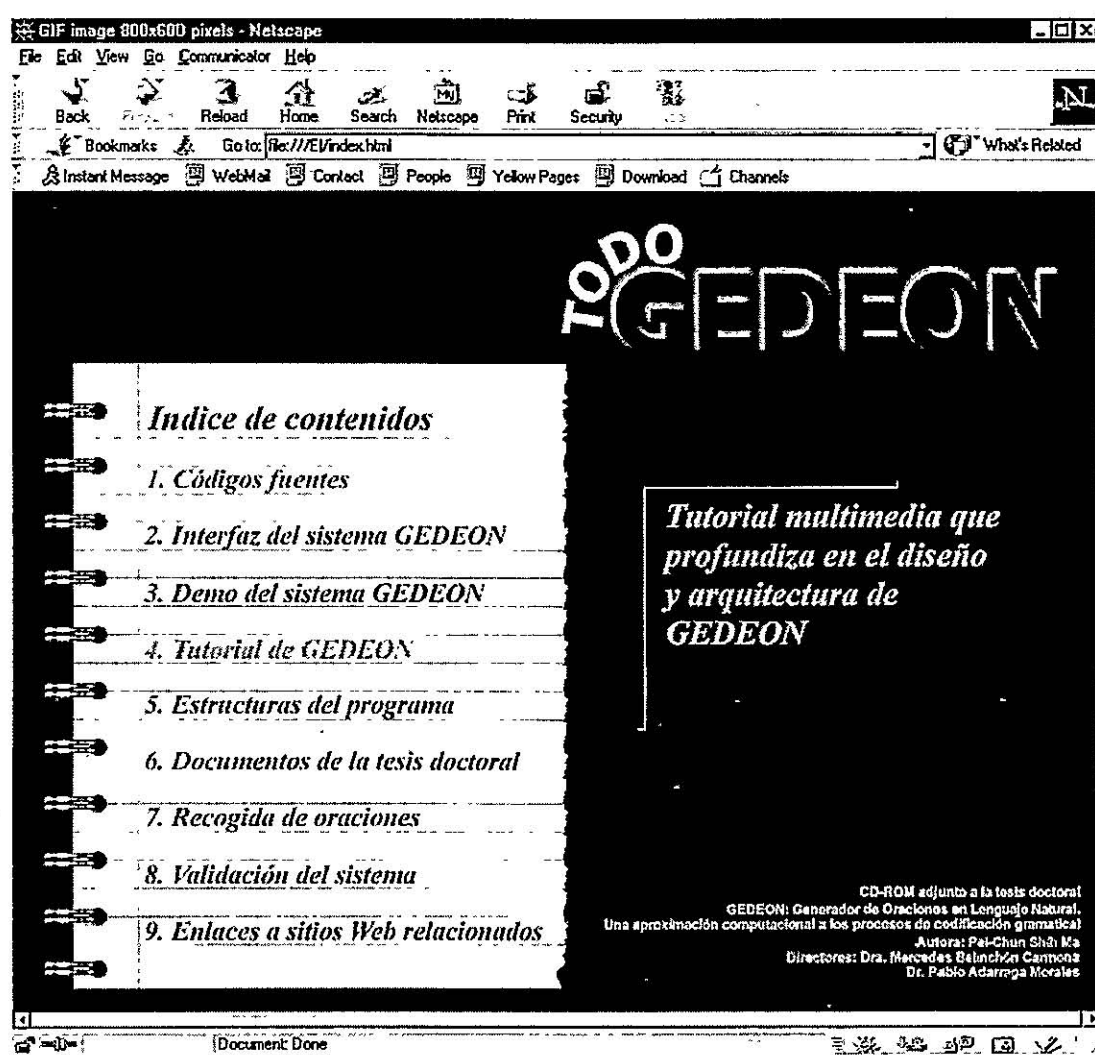


Figura 12: Pantalla Índice de Contenidos con el apartado 4 señalado.

Después de las pantallas de presentación se mostrará la página *Índice* del tutorial desde la cual se podrá acceder a cualquiera de los 4 temas que lo componen. Para ello basta con hacer clic sobre el tema deseado e inmediatamente comenzará la presentación del mismo (véase *figura 13*).

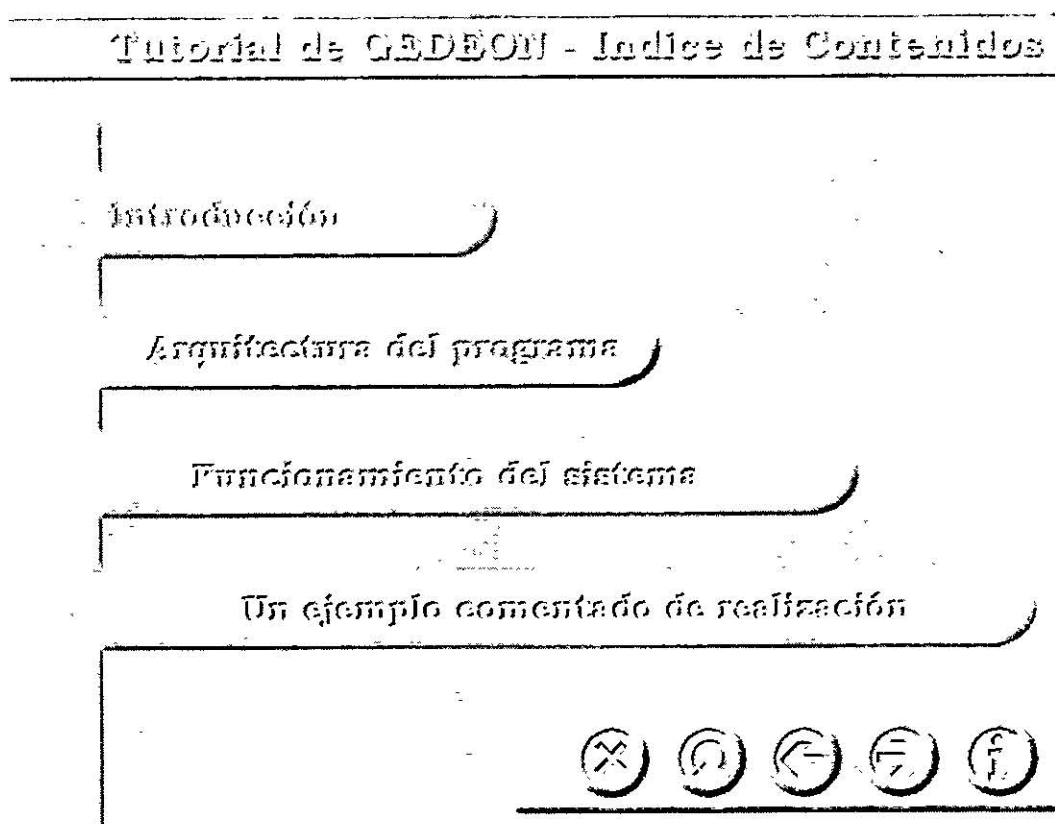


Figura 13: Pantalla de Índice de Contenidos. Tutorial de GEDEON

Como hemos comentado, éste es un entorno interactivo en el que el usuario tendrá libertad absoluta para navegar a lo largo del programa. Para ello se han creado 5 botones que se muestran en la parte inferior de todas y cada una de las pantallas del tutorial (véase *figura 14*).

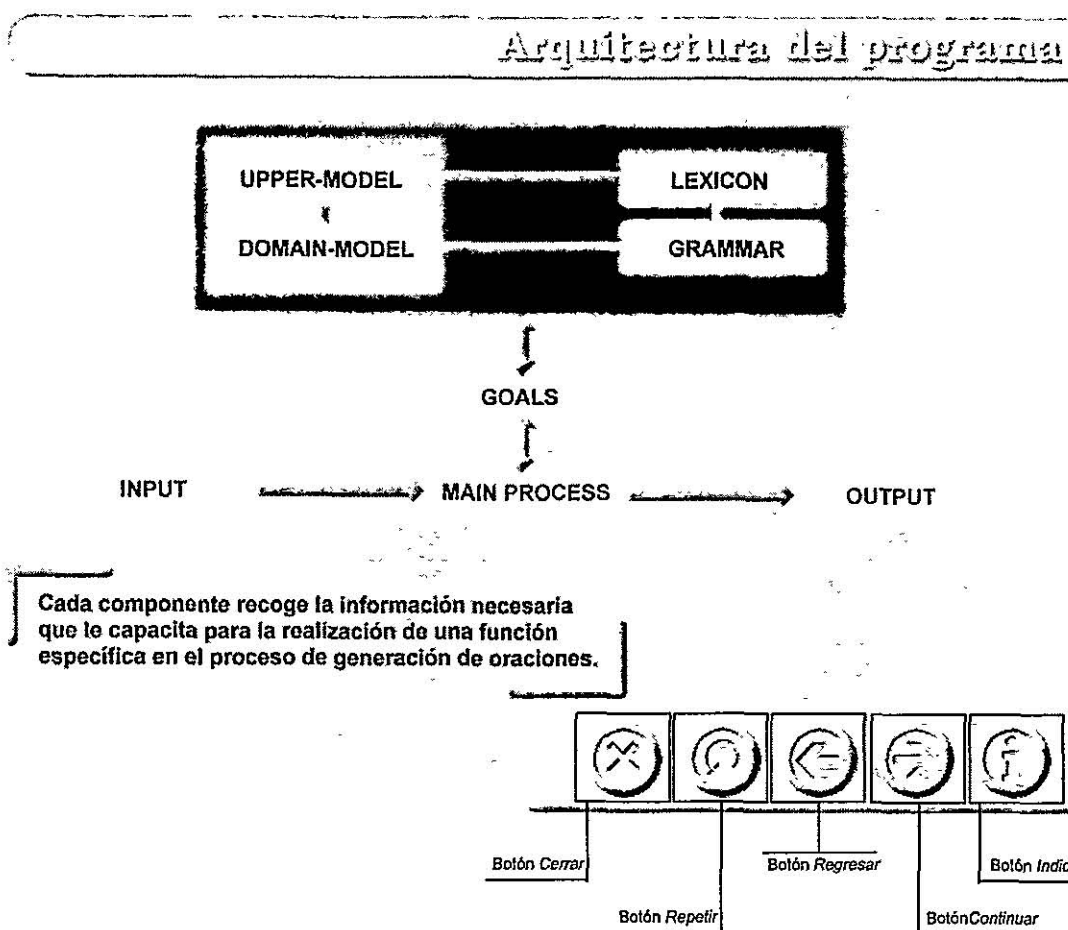


Figura 14: Barra de Botones. Tutorial de GEDEON

- Botón *Indice*: pulsando sobre este icono desde cualquier pantalla retornará a la pantalla de *Indice* del tutorial. Esto puede sernos útil si en determinado momento queremos consultar otro de los temas disponibles aún sin haber concluido el que está siendo consultado.
- Botón *Continuar*: muestra la próxima pantalla de explicación del tema que está siendo consultado.
- Botón *Regresar*: con este icono puede regresar a la pantalla anterior, útil en caso de que necesite recordar o volver a consultar algunas de las cuestiones que han sido explicadas en anteriores pantallas.
- Botón *Repetir*: cada pantalla de nuestra aplicación puede estar compuesta por textos, gráficos, animaciones y sonidos con un orden temporal importante para el entendimiento del tema. Si no ha tenido oportunidad de escuchar todas las

